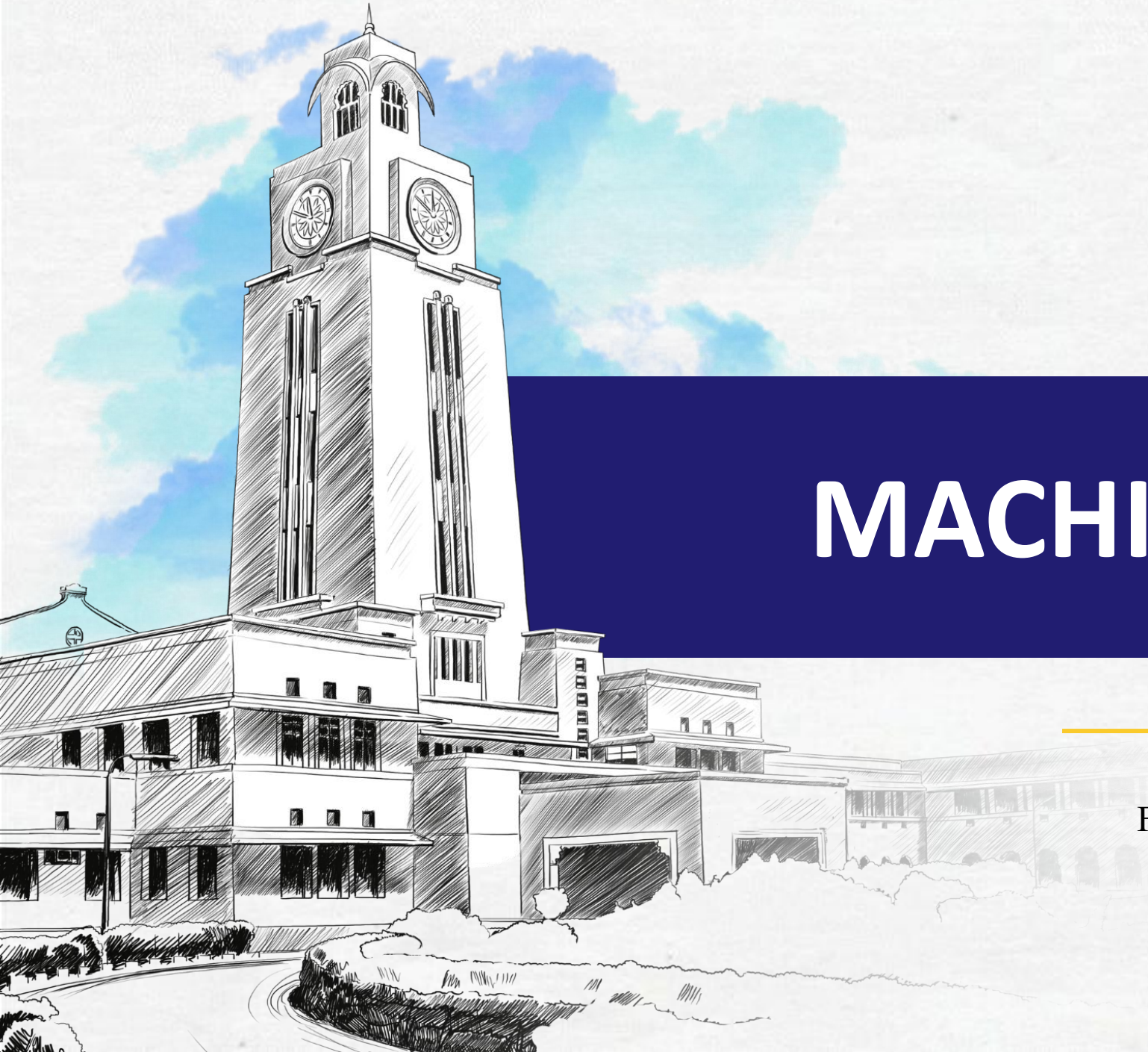


BITS Pilani
Pilani | Dubai | Goa | Hyderabad

MACHINE LEARNING

Faculty Name : Dr. A.R.Kiruthiga

Mail-Id : kiruthiga@wilp.bits-pilani.ac.in

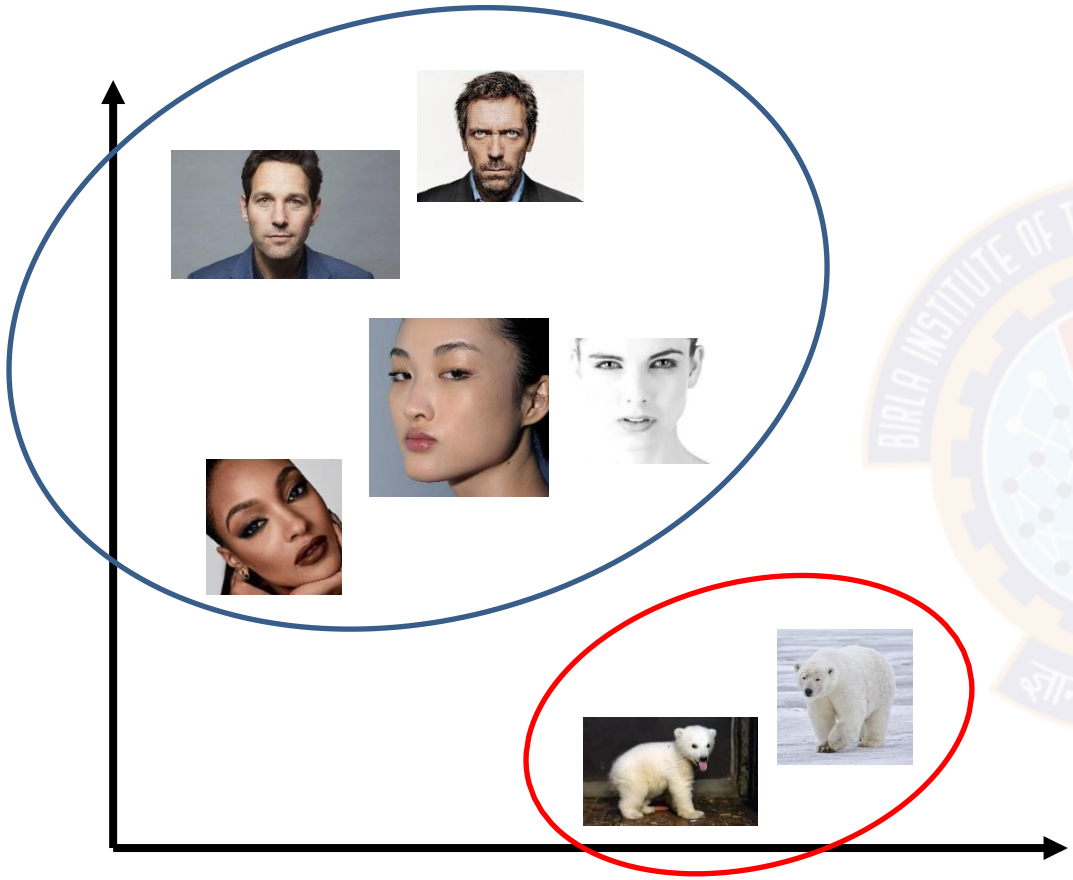
Session 15
(10th May 2026)

Unsupervised Learning

Module –Unsupervised Learning

- Mixture Models
- Expectation Maximization (EM) Algorithm
- K-means Clustering

Unsupervised discovery



Task : Group the objects based on the characterisitics features

Input:
Structured data of features like colour , shape, size, etc.,

Process: Find the similarity/dissimlarity between objects with features as the defining dimensions

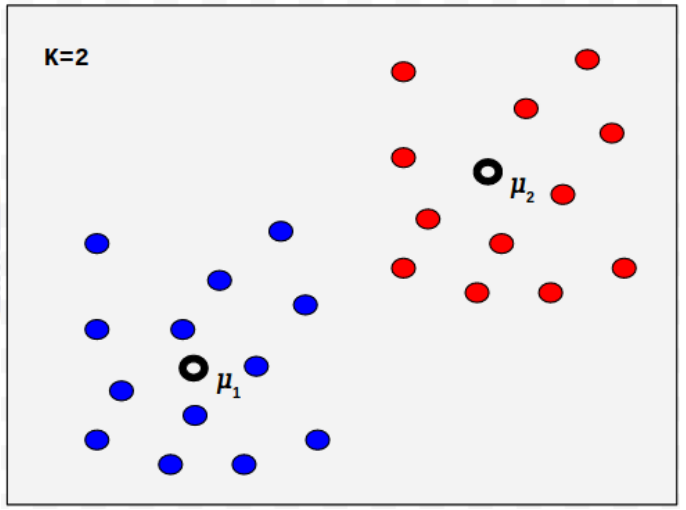
Output:
Group/Cluster of objects

K-Means Algorithm

- Works iteratively to find $\{\mu_k\}$ and $\{r_{nk}\}$ such that J is minimized

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

- Iteration involves two key steps
  - Find $\{r_{nk}\}$, fixing $\{\mu_k\}$ to minimize J
  - Find $\{\mu_k\}$, fixing $\{r_{nk}\}$ to minimize J
- Let us look at each of these steps



$$r_{nk} \in \{0, 1\}, \text{ where } k = 1, \ldots, K$$

Binary Indicator variable that indicates point xn belongs to which cluster

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

K-Means Algorithm

Initialization:

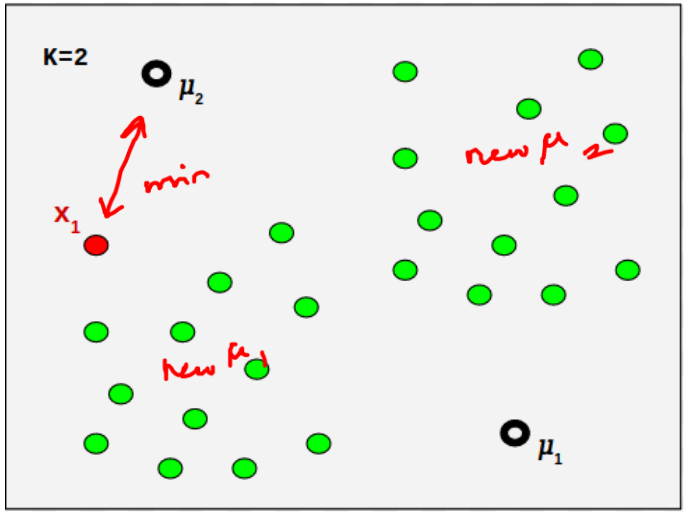
Initialize random clusters [Choose **K initial cluster centers** (also called *centroids*)].

Two - Phases:

1. Assign / Re-assign data points to clusters based on minimum distance to cluster centers
2. Re-compute the cluster means

Repeat steps 1-2 till convergence

distance < Manhattan distance, Euclidean distance



Iteration: $C1 = \{x_1, x_3, x_4, x_7 \dots x_k\}$

$C_2 = \{x_2, x_5, x_8 \dots x_j\}$

$\mu_1, \mu_2 \rightarrow$ initial cluster centers

$\|x_4 - \mu_1\|^2 \rightarrow$ minimum then assign $x_4 \in C_1$

E-step : Expectation step

Example

Suppose:

Centroid A = (2, 3) M_1

Centroid B = (8, 9) M_2

For point (3, 4): D

Distance to A is small

Distance to B is large ✓

So, during the E-step, the point is assigned to cluster A.

Mathematically:

For a point x_i,

$$c_i = \arg\min_k ||x_i - \mu_k||^2$$

Where:

- x_i =data point
- μ_k =centroid of cluster k
- c_i =assigned cluster

Euclidean distance

$$d(x,y) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

$$d(A,D) = \sqrt{(3-2)^2+(4-3)^2} = \sqrt{1^2+1^2} = \sqrt{2}$$

$$d(B,D) = \sqrt{(3-8)^2+(4-9)^2} = \sqrt{5^2+5^2} = \sqrt{50}$$

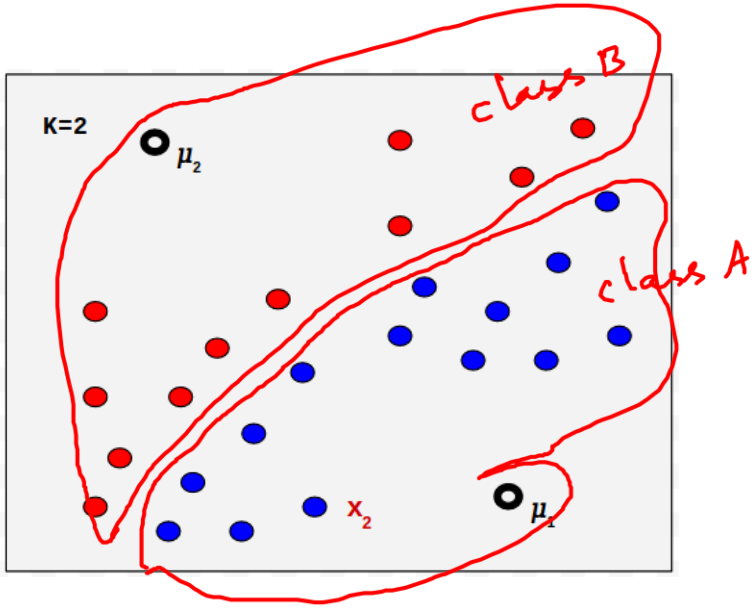
K-Means Algorithm

E-Step:

For all $x_t \in X$:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

E-step: each data point is assigned to one of the two clusters



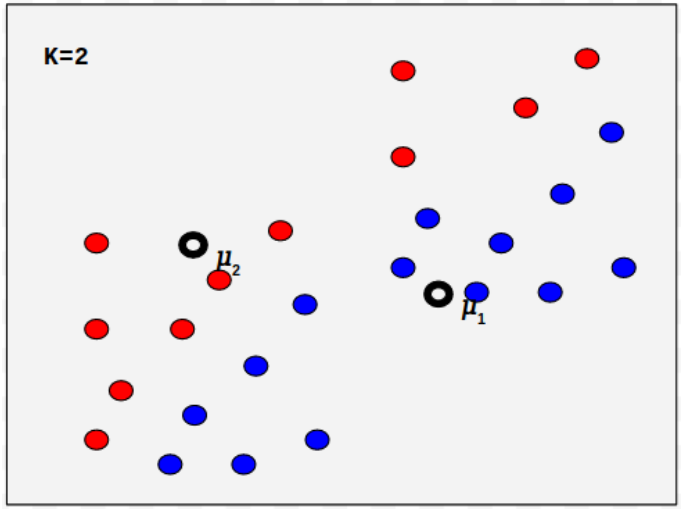
K-Means Algorithm

M-Step: Maximization step

For all μ_k[where k = 1,2,...,K] :

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

re-computing the cluster center as mean of the points assigned to the corresponding cluster.



K-Means Algorithm

Algorithm:

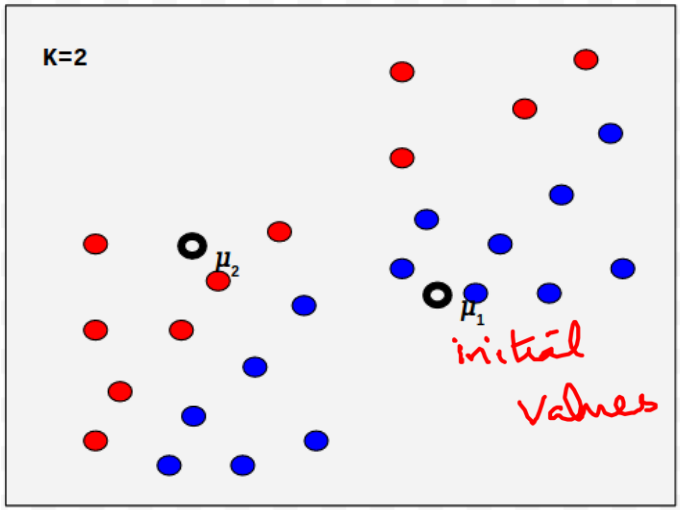
Initialize μ_k [where k = 1,2,...,K]

Repeat

E-Step [as defined earlier]

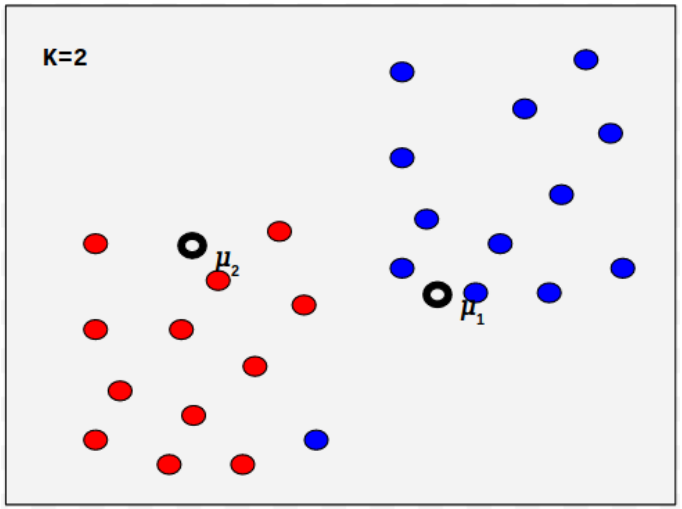
M-Step [as defined earlier]

Until convergence of μ_k .

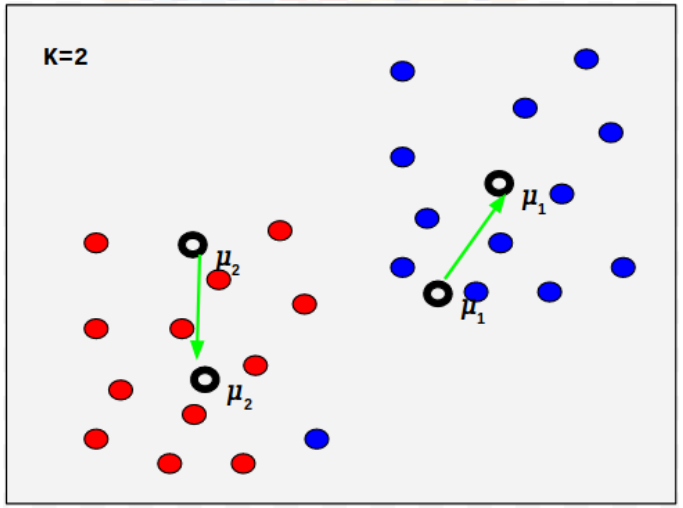


First step

initial values

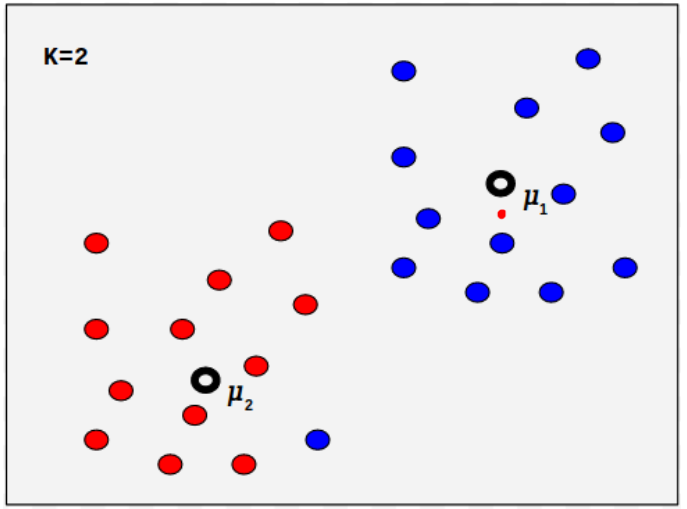


E-Step in the second iteration



M-Step in the second iteration

$|\mu_1^{new} - \mu_1| < \varepsilon$



M-Step in the second iteration

Example

Consider the analysis of weights of individuals and their respective blood glucose levels as given below:

a) Identify the clusters using K-means clustering (k=2) for the given data, assuming candidate 1 and 2 as initial centroids.

b) How many iterations does it take before termination? 2 iterations

Candidate	Weight	Glucose level	
c1 1	72	185	
c2 2	56	170	
3	60	168	c2
4	68	179	c1
5	72	182	c1
6	77	188	c1
7	70	180	c1
8	84	183	c1

Example

Candidate	Cluster1(72, 185)	Cluster2(56, 170)
3(60, 168)	$=17^2+12^2$	$=2^2+4^2$
4(68 179)	$=6^2+4^2$	$=9^2+12^2$
5(72, 182)	$=3^2+0^2$	$=12^2+16^2$
6(77, 188)	$=3^2+5^2$	$=18^2+11^2$
7(70, 180)	$=5^2+2^2$	$=10^2+14^2$
8(183, 84)	$=2^2+12^2$	$=13^2+28^2$

After 1st iteration, the cluster groups are C1{1, 4, 5, 6, 7, 8} and C2{2, 3}

- **Re-computing centroids**:

C1([72+68+72+77+70+84]/6, [185 +179+182+188+180+183]/6) =C1(73.83, 182.83)

C2([56 +60]/2, [170+168]/2) =C2(58, 169)

- **2nd Iteration** –computing distances from resulting centroids:

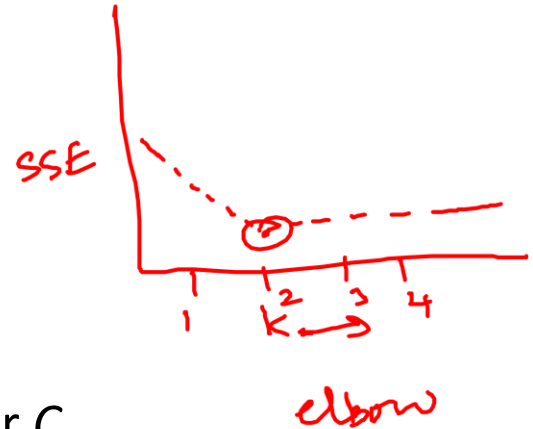
After 2nd iteration, the cluster groups are C1{1, 4, 5, 6, 7, 8} and C2{2, 3}; there is no change in cluster groups hence the algorithm terminates.

Candidate	Cluster1(182.83, 73.83)	Cluster2(169, 58)
1(185, 72)	$(182.83-185)^2+(73.83-72)^2=2.17^2+1.83^2$	$(169-185)^2+(58-72)^2=16^2+14^2$
2(170, 56)	$(182.83-170)^2+(73.83-56)^2=12.83^2+17.83^2$	$(169-170)^2+(58-56)^2=1^2+2^2$
3(168, 60)	$(182.83-168)^2+(73.83-60)^2=14.83^2+13.83^2$	$(169-168)^2+(58-60)^2=1^2+2^2$
4(179, 68)	$(182.83-179)^2+(73.83-68)^2=3.83^2+5.83^2$	$(169-179)^2+(58-68)^2=10^2+10^2$
5(182, 72)	$(182.83-182)^2+(73.83-72)^2=0.83^2+1.83^2$	$(169-182)^2+(58-72)^2=13^2+14^2$
6(188, 77)	$(182.83-188)^2+(73.83-77)^2=5.17^2+3.17^2$	$(169-188)^2+(58-77)^2=19^2+19^2$
7(180, 70)	$(182.83-180)^2+(73.83-70)^2=2.17^2+1.83^2$	$(169-180)^2+(58-70)^2=11^2+12^2$
8(183, 84)	$(182.83-183)^2+(73.83-84)^2=0.17^2+10.17^2$	$(169-183)^2+(58-84)^2=14^2+26^2$

Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

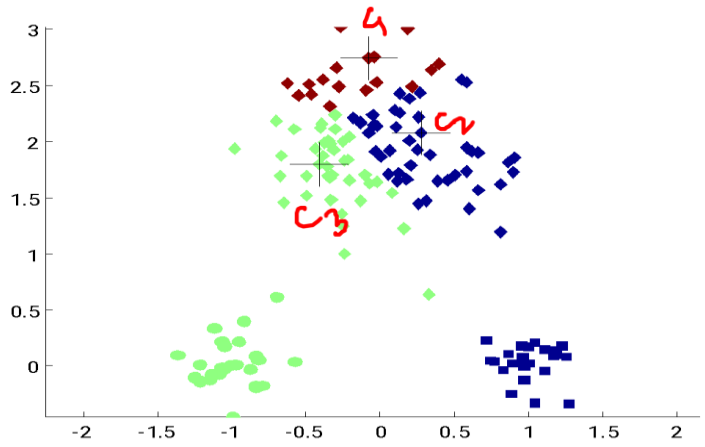
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

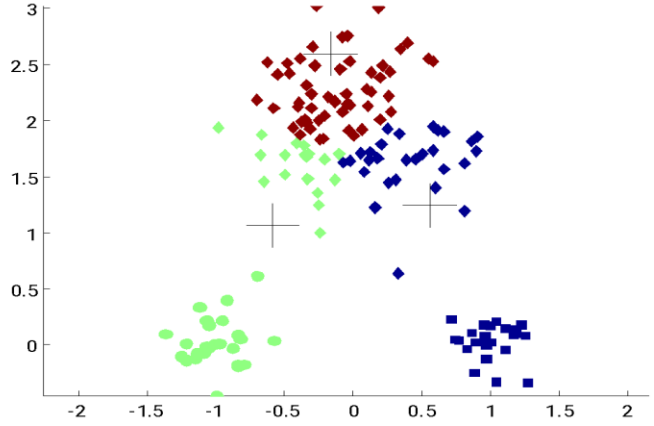


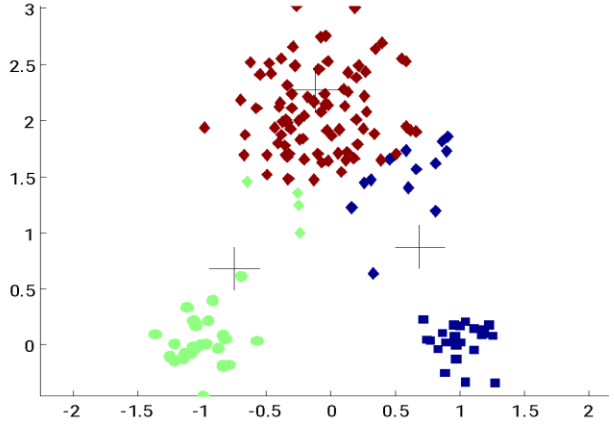
Candidate 1

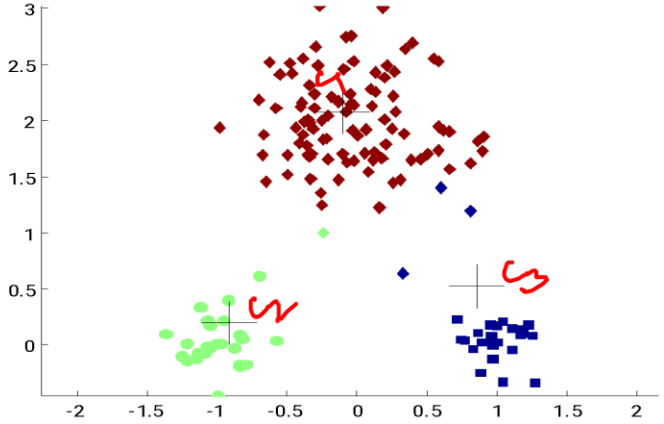
  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

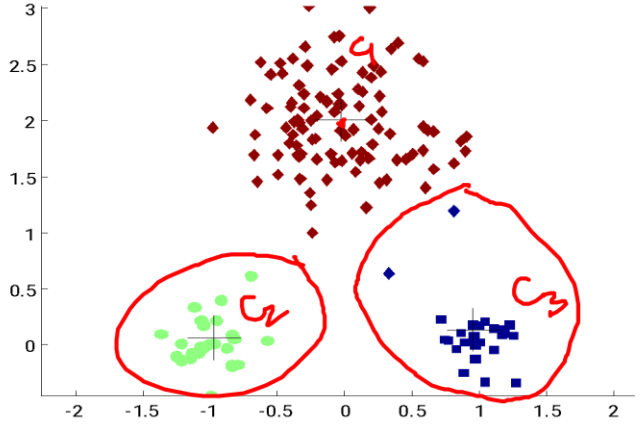
Importance of Choosing Initial Centroids

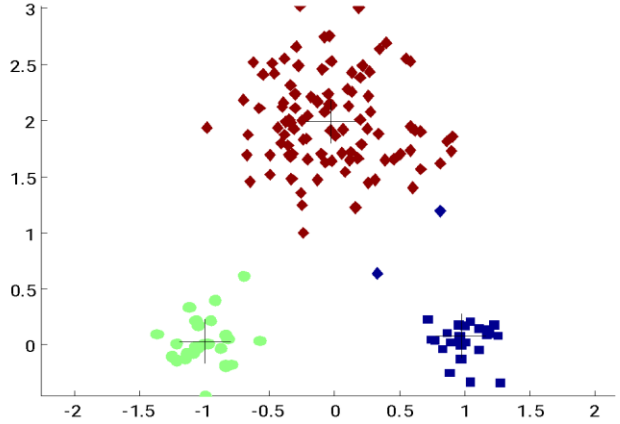












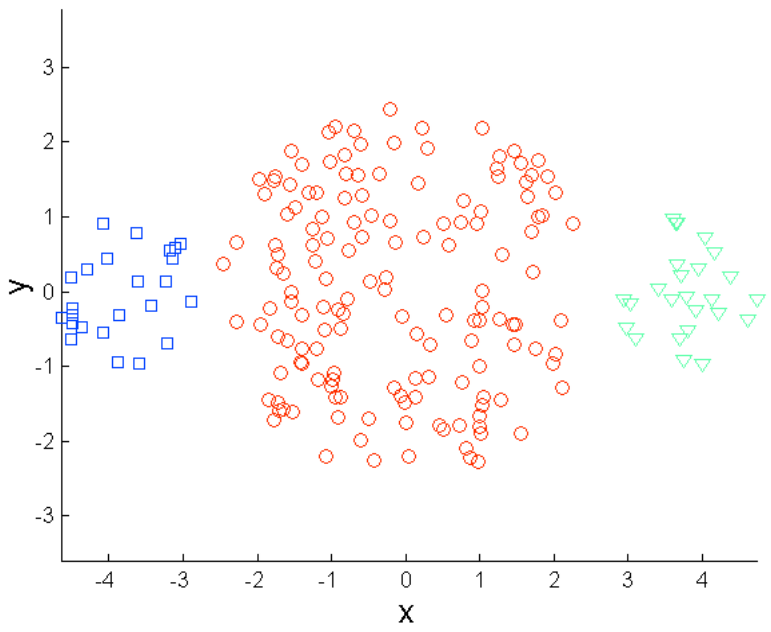
Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.

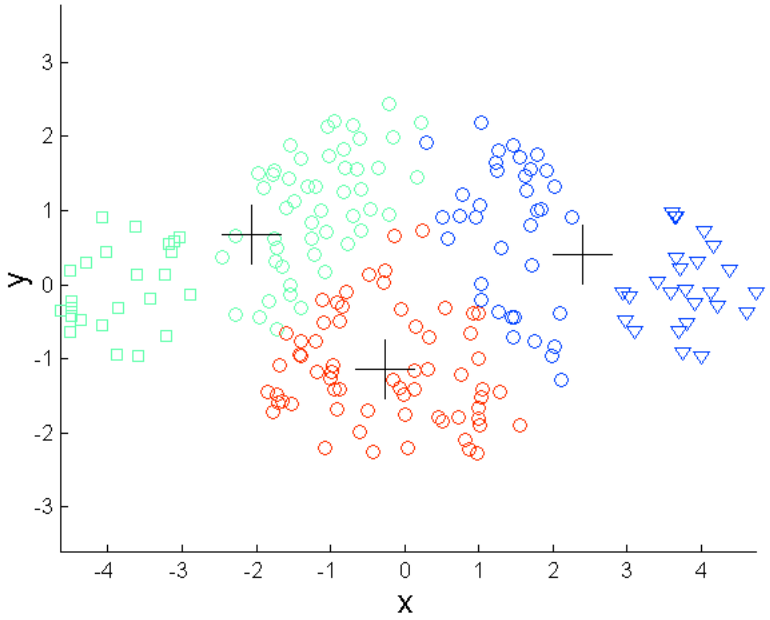
0 1, 2. . . 10, 1000

as it affects the centroid of the cluster.

Limitations of K-means: Differing Sizes

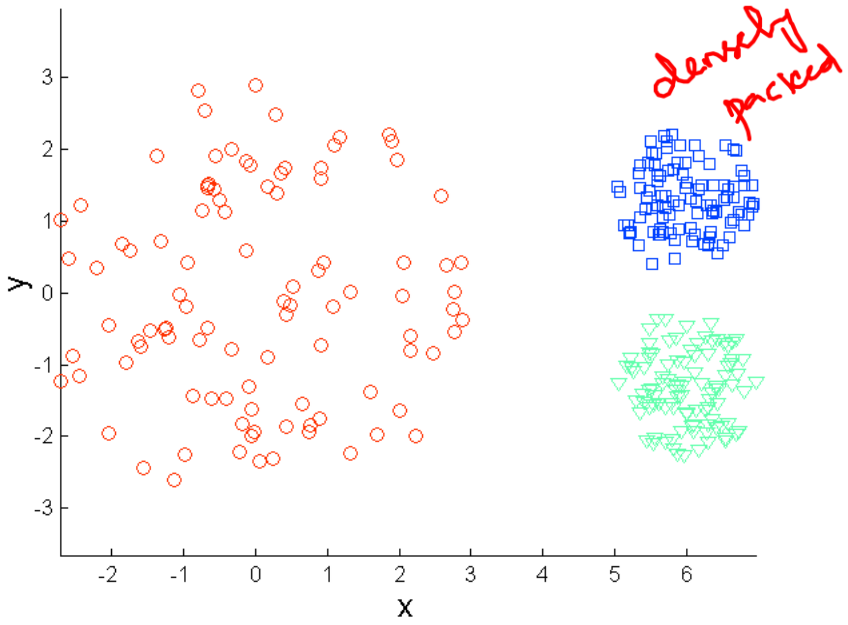


Original Points

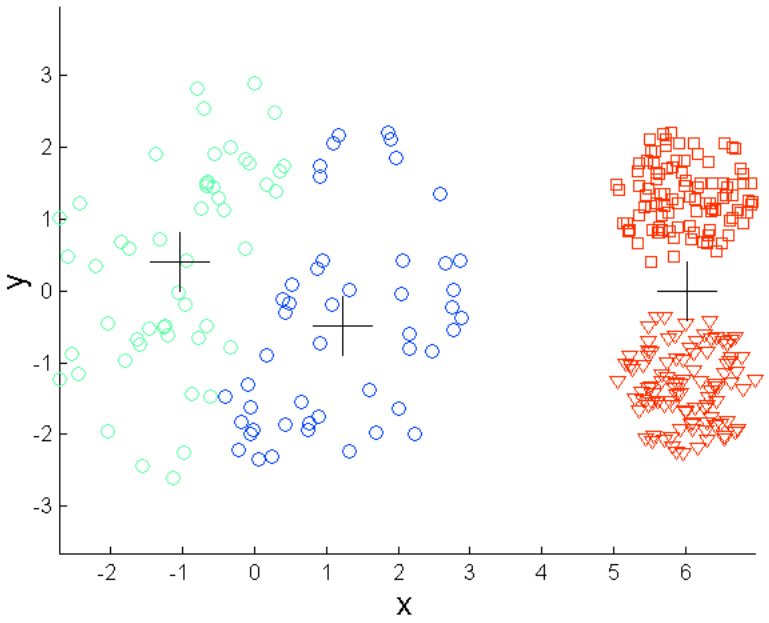


K-means (3 Clusters)

Limitations of K-means: Differing Density

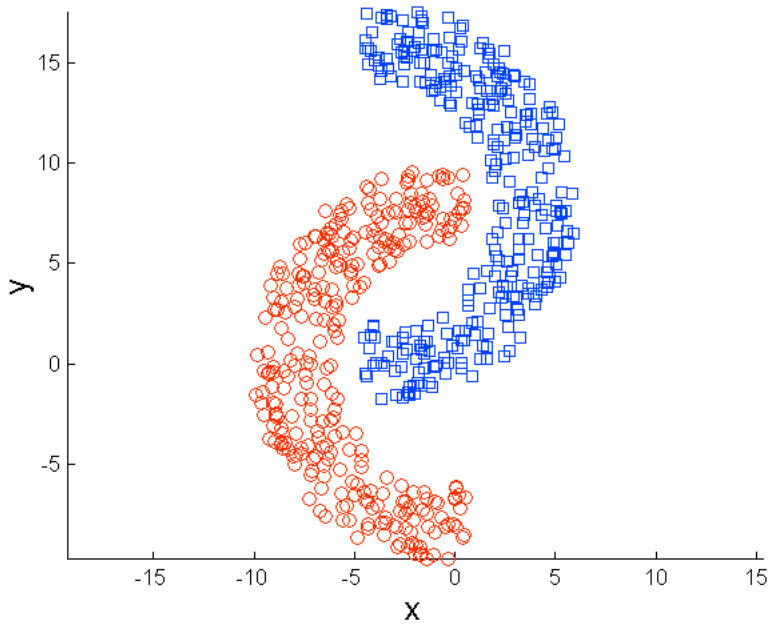


Original Points

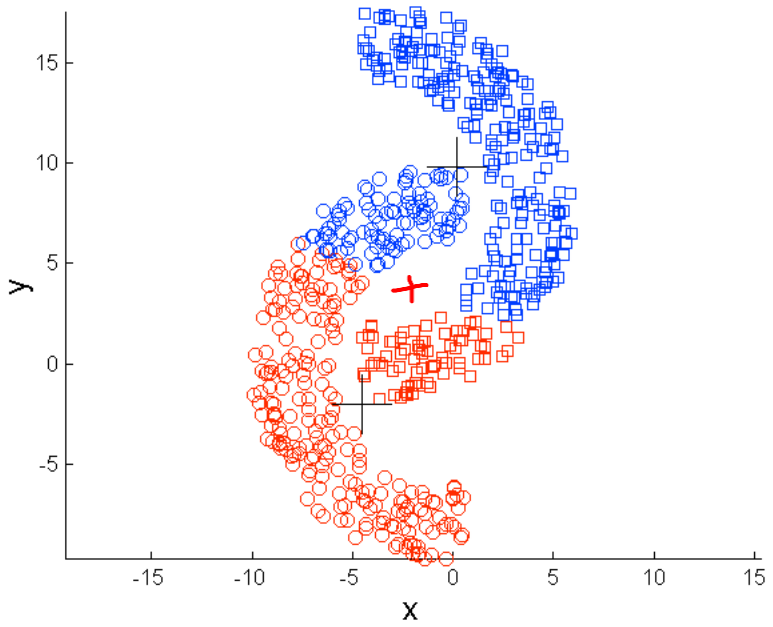


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

From K-Means to K-Medoids

- The k-means is sensitive to outliers-since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.
- **K-Medoids**

1. Initialize: select k random points out of the n data points as the medoids.

2. Associate each data point to the closest medoid by using any common distance metric methods.

3. While the cost decreases:

For each medoid m, for each data o point which is not a medoid:

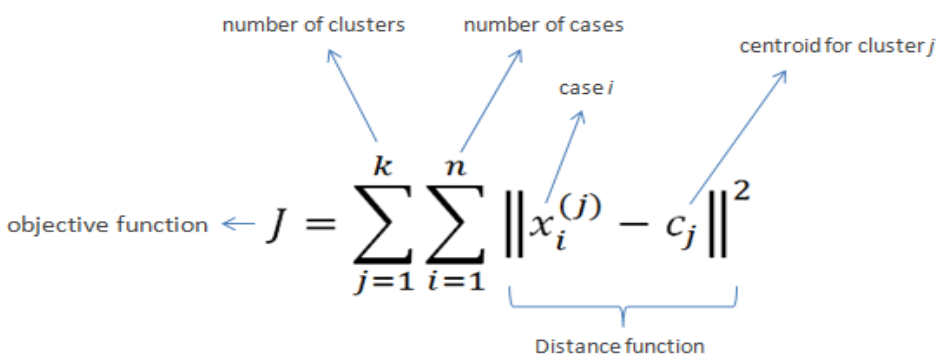
1. Swap m and o, associate each data point to the closest medoid, recompute the cost.

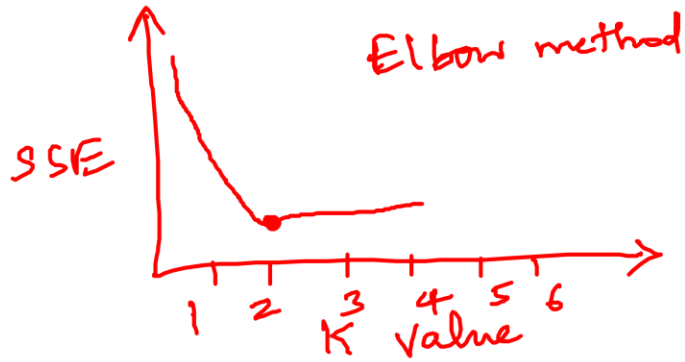
2. If the total cost is more than that in the previous step, undo the swap.

*instead of using the **mean (centroid)** of a cluster, it uses an actual **data point** as the center, called a **medoid**.

How to choose a “good” k for k-means clustering?

- Choose the number of clusters by visually inspecting data points
- **Elbow method:**
  - Compute the sum of squared error (SSE) for some values of k (for example, 2,3,4,5,6…). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid.



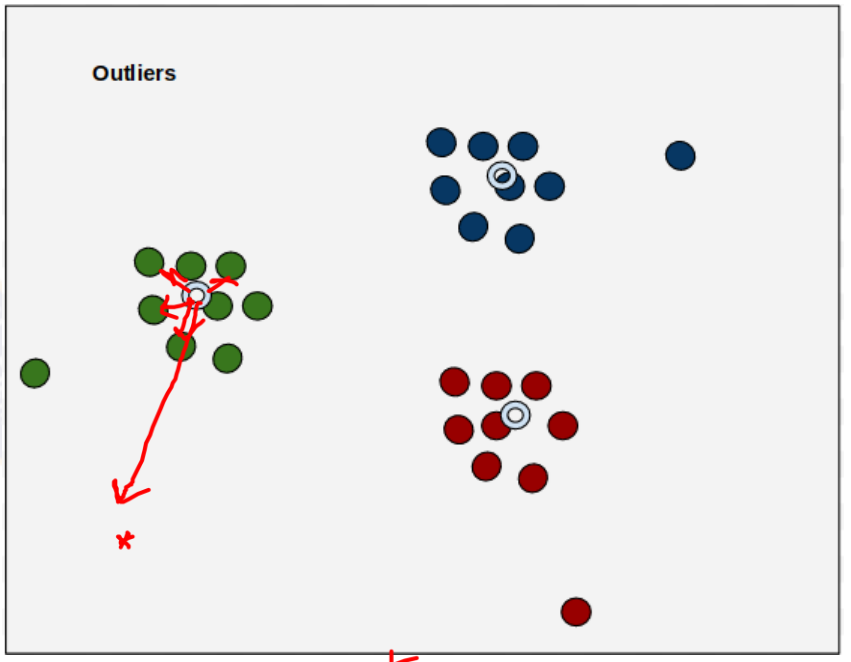


  - If you plot k against SSE, you will see that the error decreases as k gets larger, this is because when the number of clusters increases, they should be smaller, so distortion is also smaller. The idea of the elbow method is to choose the k at which the SSE decreases abruptly.

(naive) K-Means for detecting outliers

- Let dist(x, μ_k) be the distance of a point x, assigned to cluster k to its center μ_k.
  - $L\mu_k$ be the the average distance of all the points assigned to cluster k with its center

- The ratio dist(x, μ_k) / $L\mu_k$ for each point is the outlier score for each point.
  - Higher the ratio for a point x, more likely x is an outlier



$d(x_1, M_K)$ ratio

x_1 25 $25/100 = 0.25$

x_2 15 $15/100 = 0.15$

x_{10} 1000

$dist(x_{10}, M_K)$ = greater value = 1000 (for eg)

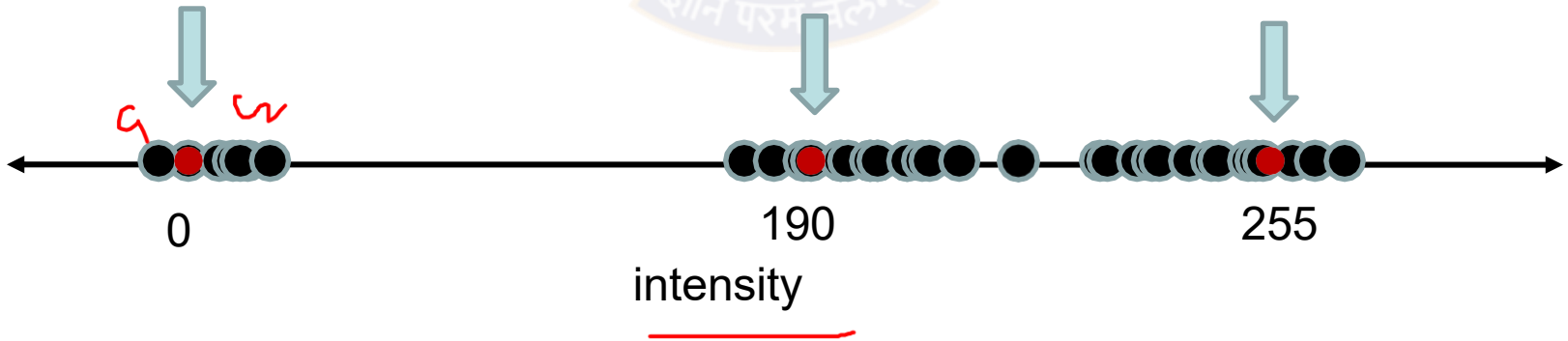
$LM_K = 100$

$ratio = \frac{1000}{100} = 10$ ✓

Notion of Mixture Models

K means – Hard clustering

- Assigned each example to exactly one cluster
- What if clusters are overlapping?
  - Hard to tell which cluster is right
  - Maybe we should try to remain uncertain
- What if cluster has a non-circular shape?
- Hard assignments of data points to clusters – small shift of a data point can flip it to a different cluster
- For example, if a point is near the 'border' between two clusters, it's often better to know that it has near equal membership probabilities for these clusters, rather than blindly assigning it to the nearest one.



1 of K coding mechanism

- For each data point x_n, we introduce a set of binary indicator variables $r_{nk} \in [0,1]$ such that

$$\sum_k r_{nk} = 1$$

where k =1,...,K describing which of the K clusters the data point x_n is assigned to, so that if data point x_n is assigned to cluster k then $r_{nk} > r_{nj}$ for j not equal to k.

- Example: 5 data points and 3 clusters

$$(r_{nk}) = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{array} \begin{array}{c} \begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

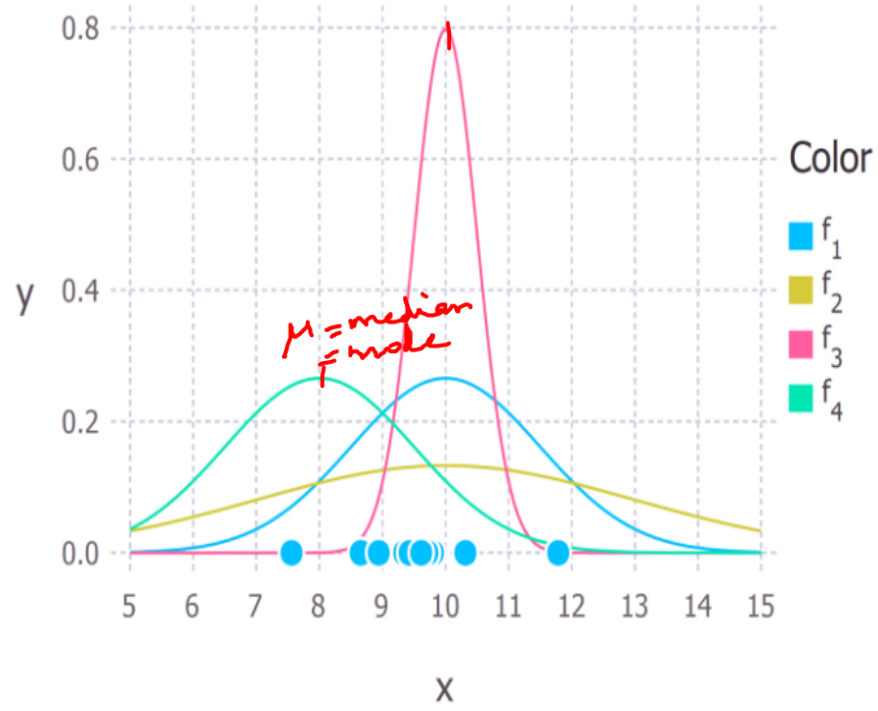
GMM - soft Clustering

- **Solution**: replace 'hard' clustering of K-means with 'soft' probabilistic assignments
- Probabilistic Clustering
  - Represents the probability distribution of the data as a **Gaussian mixture model (GMM)**
  - GMMs give a probabilistic assignment of points to clusters which quantify uncertainty.
  - Mixture models combines a set of distributions to create a convex space where we can search for the optimal parameters for such distributions using MLE.
    - Mixture models in general don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning.
- In GMM, Clusters modeled as Gaussians
- **EM algorithm:** assign data to cluster with some probability

Gaussian Distribution

- The normal curve is *bell-shaped* and has a single peak at the exact center of the distribution.
- The arithmetic mean, median, and mode of the distribution are equal and located at the peak.
- Half the area under the curve is above the peak, and the other half is below it.
- The normal distribution is symmetrical about its mean.

Holds	Fails
Amalgamation of similar distributions	Dominated by one (or few) particular distribution
Contributing factors are independent	Dependencies among contributing factors
Sample selection is uniformly random	Sample selection is correlated to previous selection



The 10 data points and possible Gaussian distributions from which the data were drawn. f1 is normally distributed with mean 10 and variance 2.25 (variance is equal to the square of the standard deviation), this is also denoted f1 ~ N (10, 2.25). f2 ~ N (10, 9), f3 ~ N (10, 0.25) and f4 ~ N (8, 2.25)

MATHEMATICAL FUNCTION (Pdf)

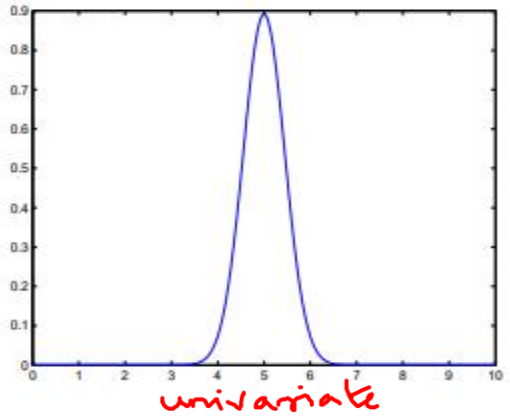
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Gaussian / Normal distribution

This is a bell shaped curve with different centers and spreads depending on μ and σ

Note constants:
π=3.14159
e=2.71828

Univariate and Multivariate Gaussian density



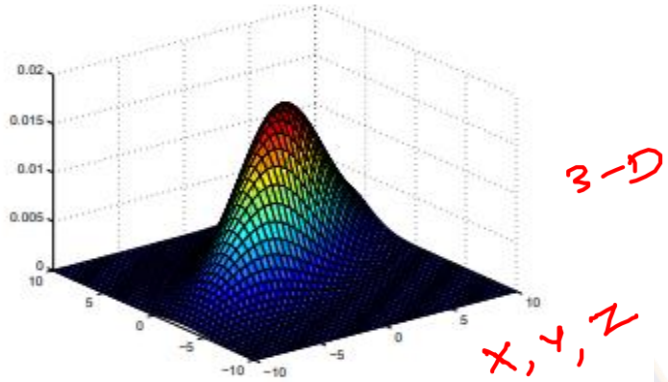


Figure 1: The figure on the left shows a univariate Gaussian density for a single variable X. The figure on the right shows a multivariate Gaussian density over two variables X_1 and X_2.

If $\mathbf{x}$ is multivariate normal, it also has a **mean vector** μ such that

$$\mu = (\mathbb{E}(X_1), \mathbb{E}(X_2), \ldots, \mathbb{E}(X_n))$$

where $\mathbb{E}(X)$ is the expected value (or mean) of X. $\mathbf{x}$ also has a **covariance matrix** Σ satisfying

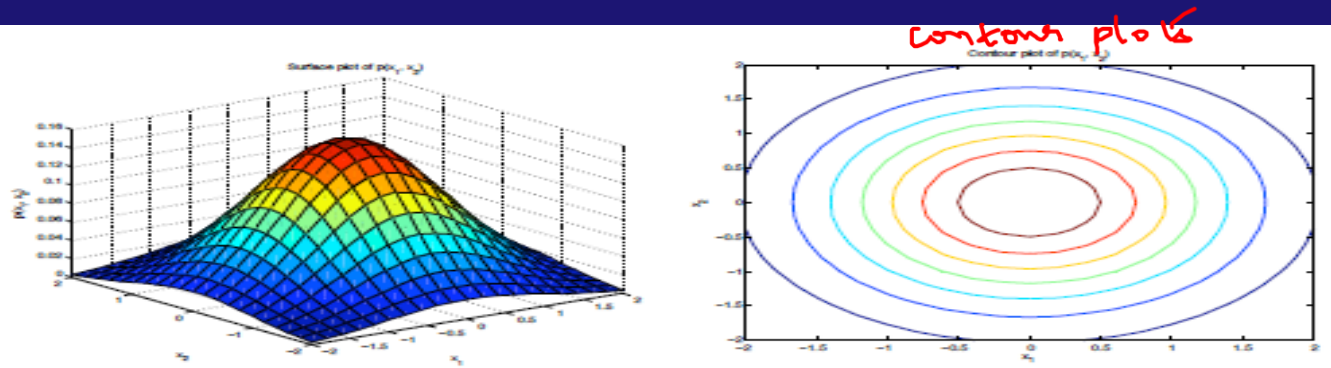
$$\Sigma_{i,j} = \mathrm{Cov}(X_i, X_j)$$

where $\mathrm{Cov}(X_i, X_j)$ is the covariance of X_i and X_j.

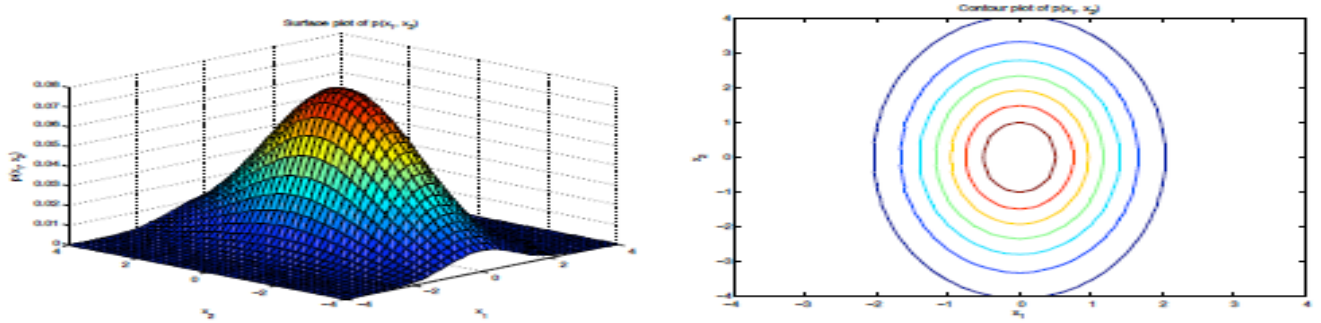
Furthermore, $\mathbf{x}$ is completely defined by μ and Σ, so it is convenient to write

$$\mathbf{x} \sim \mathcal{N}(\mu, \Sigma).$$

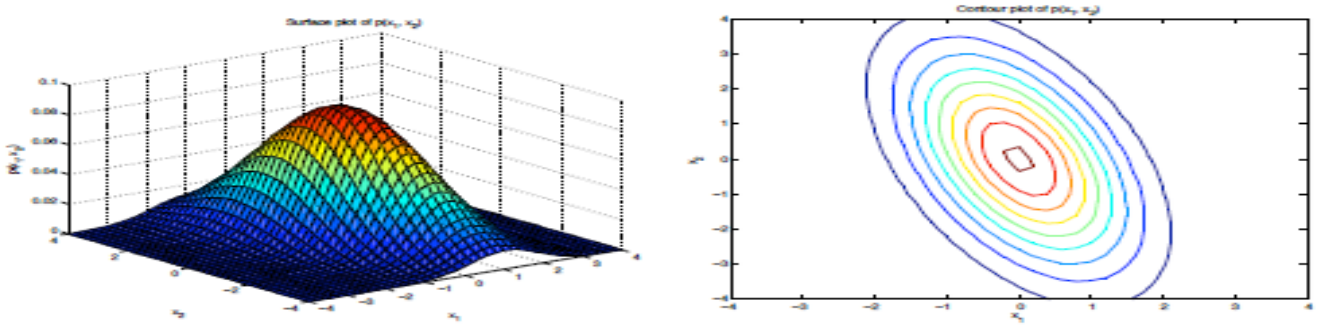
2-dimensional Gaussian distribution



(a) Spherical Gaussian (diagonal covariance, equal variances)



(b) Gaussian with diagonal covariance matrix



(c) Gaussian with full covariance matrix

Figure 7: Surface and contour plots of 2–dimensional Gaussian with different covariance structures

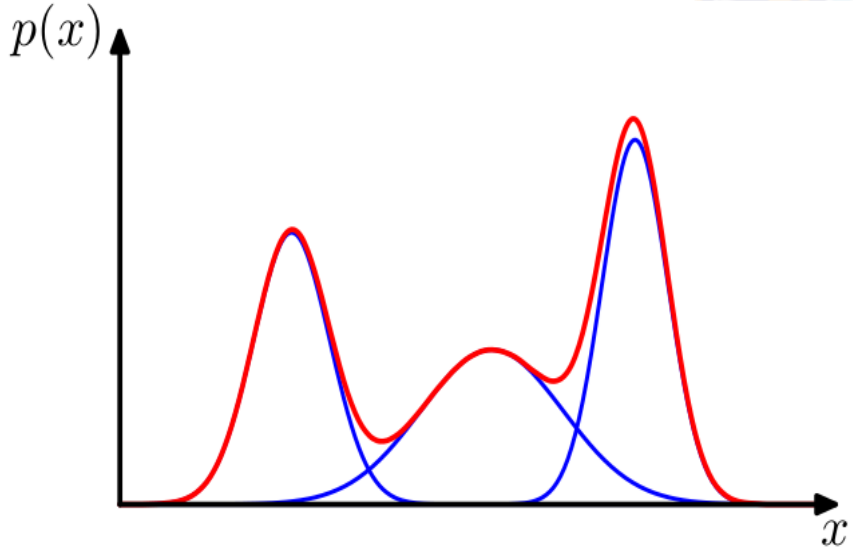
A diagonal covariance matrix means that the variables are uncorrelated, and the matrix only has values on its diagonal.

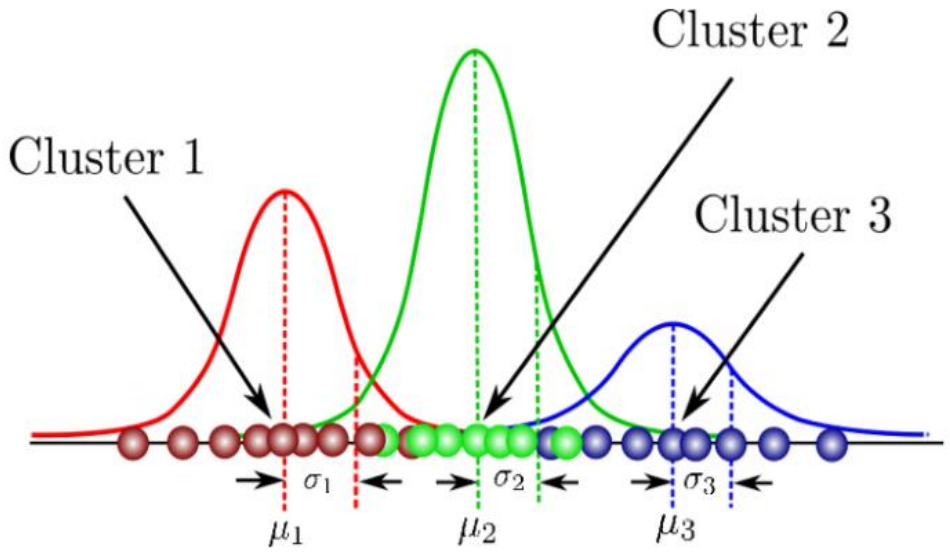
Unequal variances: there is still no correlation between the two dimensions

Full covariance: correlation between the dimensions

GMM – Mixture of Gaussians

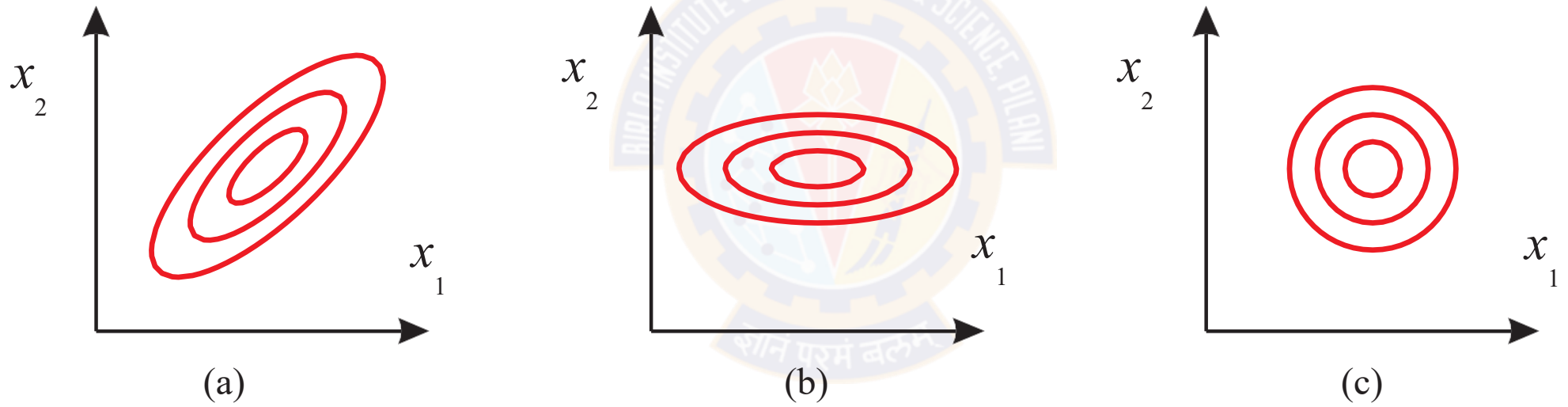
- Probabilistic cluster is a distribution over the data space, which can be mathematically represented using a probability density function (or distribution function).
- Our task: infer a set of k probabilistic clusters that is mostly likely to generate D





Example: Mixture of 3 Gaussians

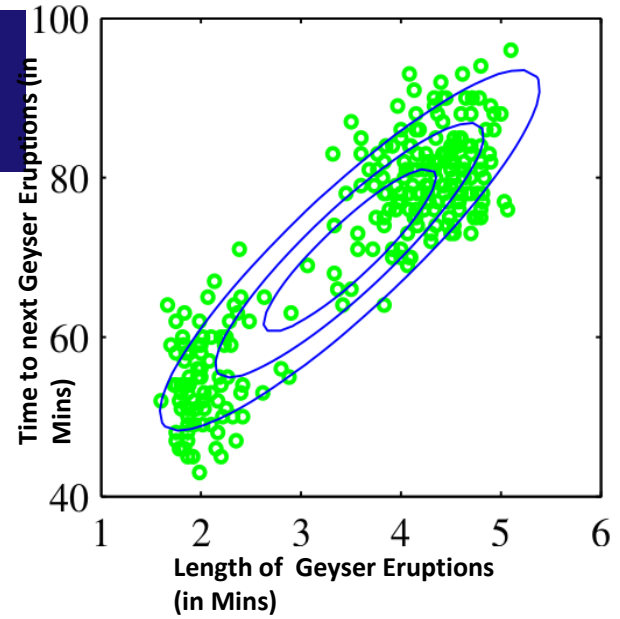
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

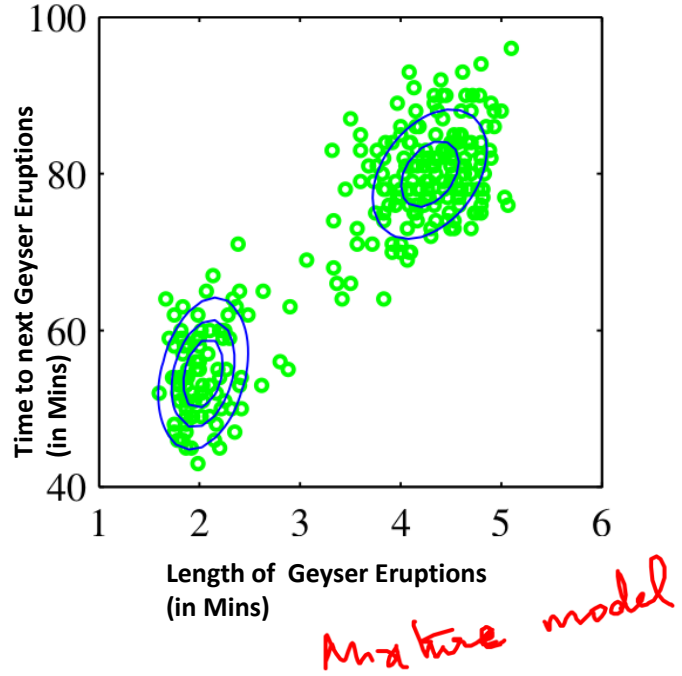


- The observed data come from a Gaussian mixture distribution which consists of K Gaussians with their own means and variances.
- K classes are latent

GMM – Why single distribution is insufficient

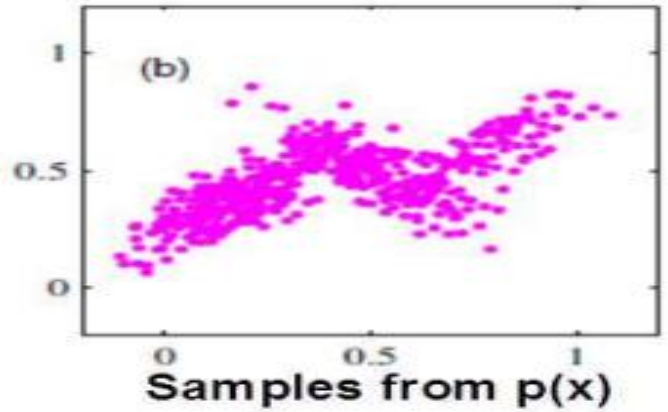
- Gaussian distribution is that it is intrinsically unimodal
- A single distribution is not sufficient to fit many real data sets which is multimodal
- A mixture model assumes that a set of observed objects is a mixture of instances from multiple probabilistic clusters, and conceptually each observed object is generated independently





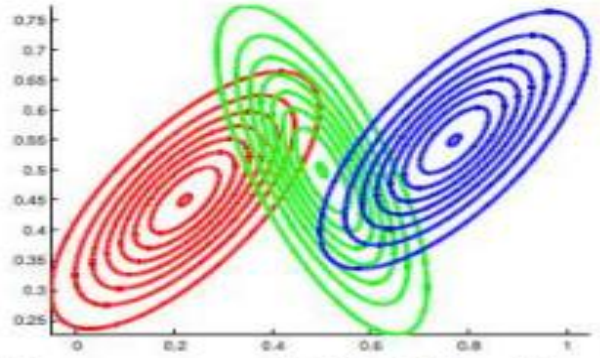
Learning a Mixture of Gaussians

Our actual observations

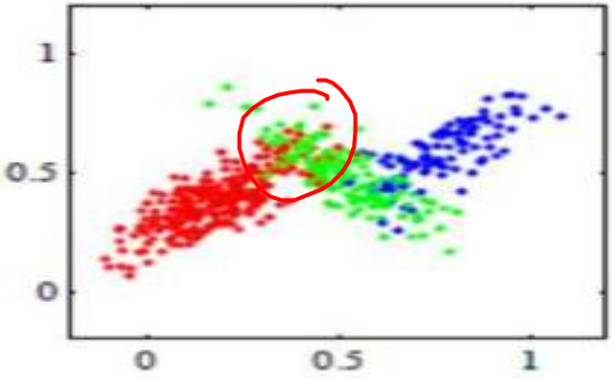


Samples from p(x)

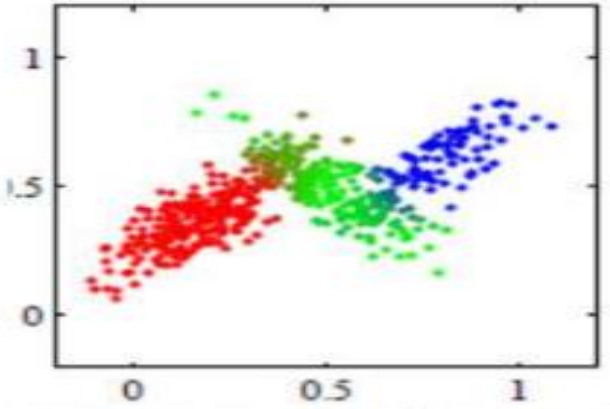
Mixture of 3 Gaussians



Component Distributions

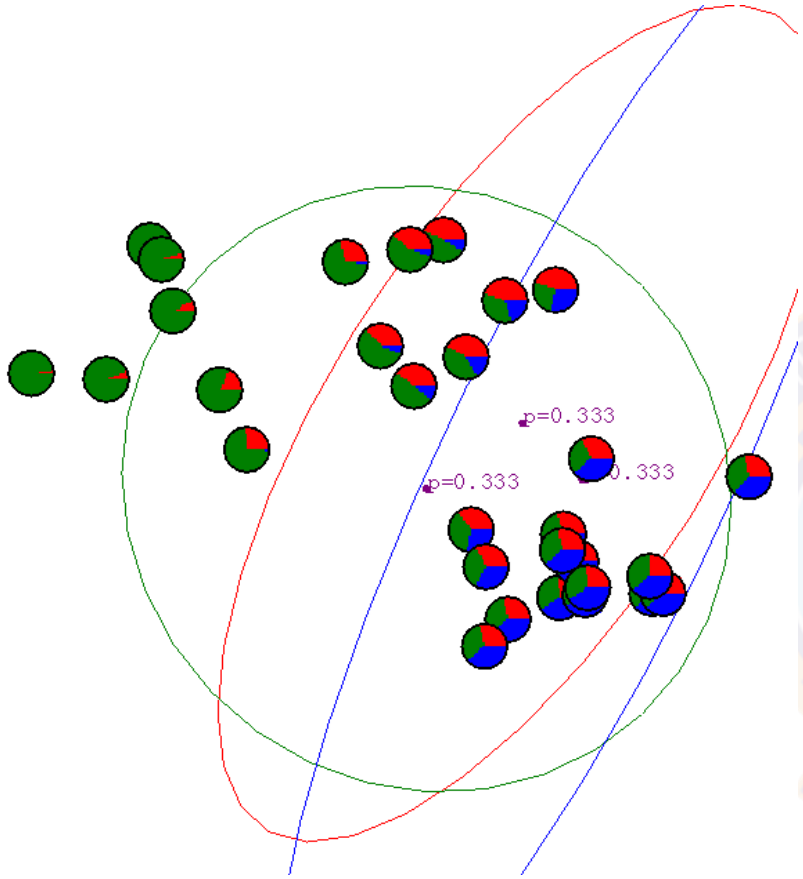


Samples labeled using their true component



Soft clustering learned by a Gaussian mixture model

Gaussian Mixture Example: Start

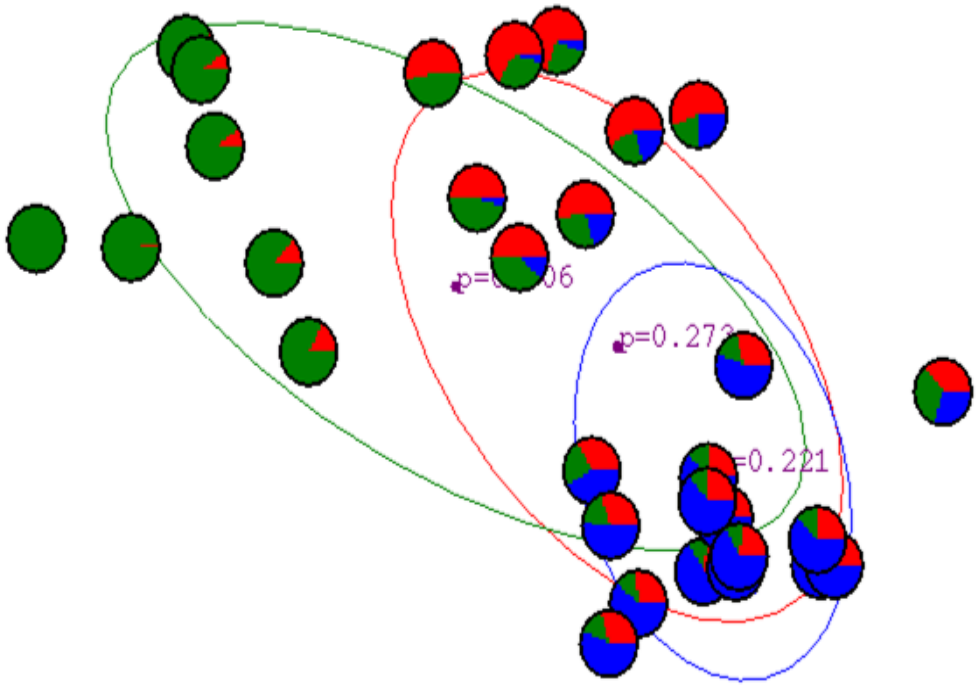


$$P(y = \bullet \mid x_j, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3, p_1, p_2, p_3)$$

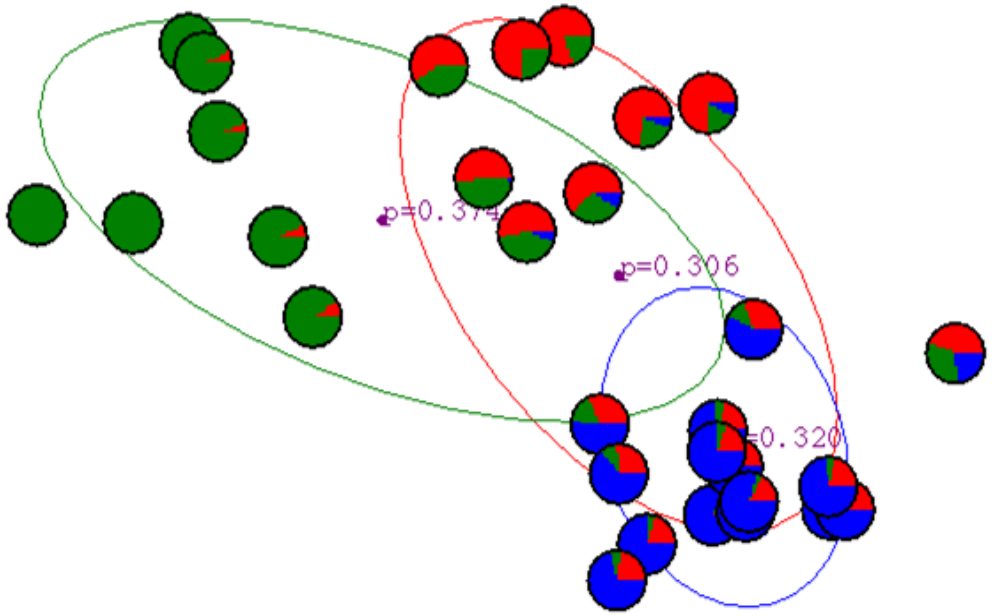
mean values

covariances

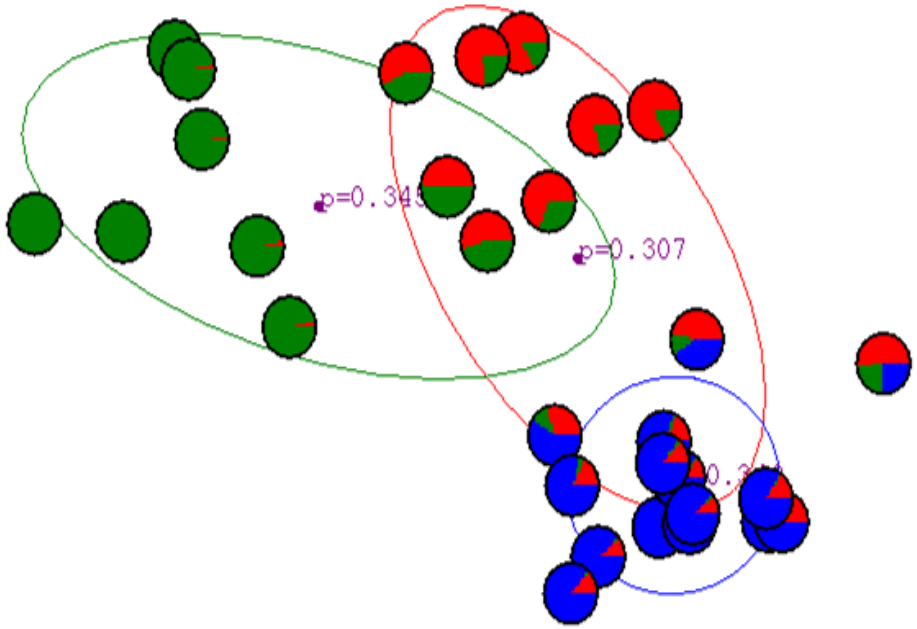
After first iteration



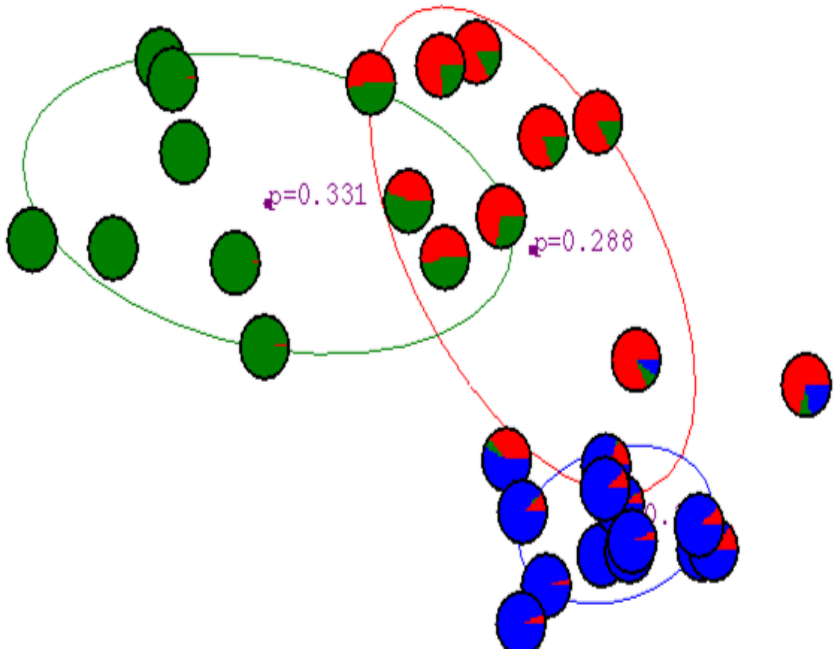
After 2nd iteration



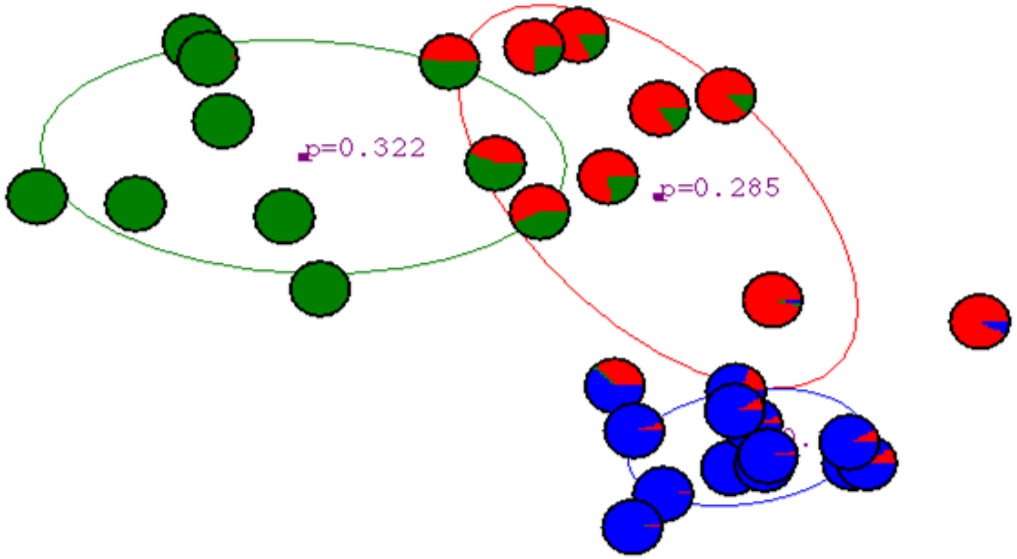
After 3rd iteration



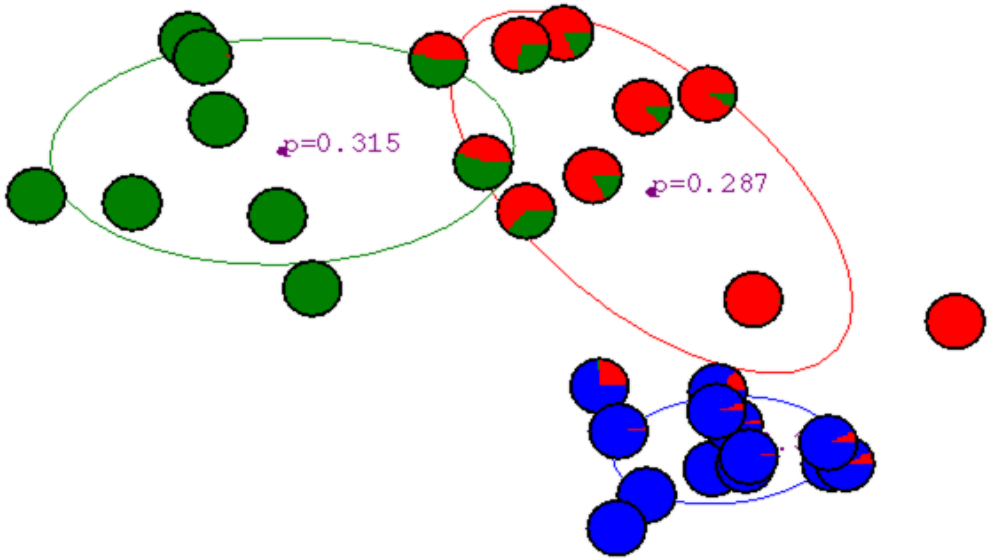
After 4th iteration



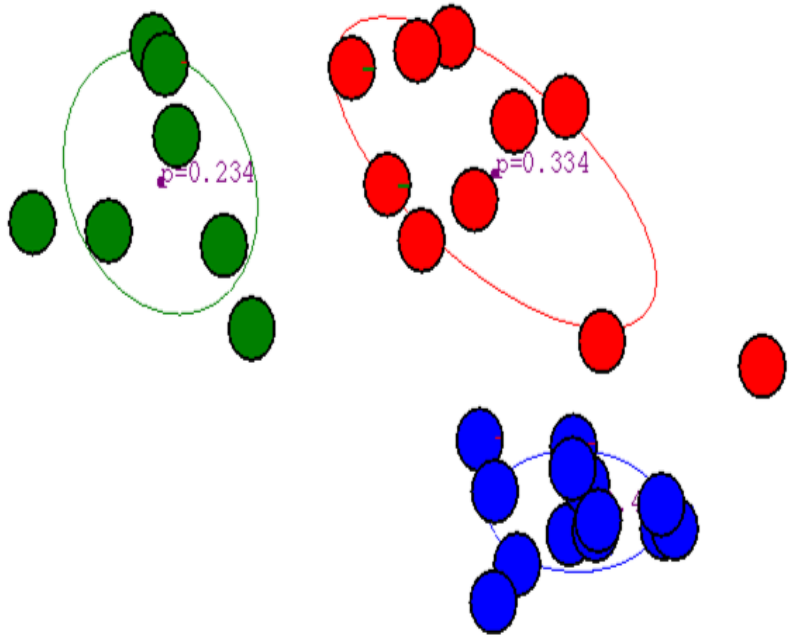
After 5th iteration



After 6th iteration



After 20th iteration



GMM equations

- **Mixture models**

K = number of mixture components (clusters),

π_j = mixture weights

$p_j(x)$ = family of distributions

(Gaussian, Bernoulli, etc).

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$$

$$p(\mathbf{x}) = \sum_{j=1}^{K} \pi_j p_j(\mathbf{x})$$

$$0 \leq \pi_j \leq 1, \quad \sum_{j=1}^{K} \pi_j = 1$$

- **Gaussian Mixture models**

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})$$

$$0 \leq \pi_j \leq 1, \quad \sum_{j=1}^{K} \pi_j = 1$$

θ = collection of all the parameters of the model (mixture weights, means, and covariance matrices):

$$\boldsymbol{\theta} := \{\pi_1, \cdots, \pi_K, \boldsymbol{\mu_1}, \cdots, \boldsymbol{\mu_K}, \boldsymbol{\Sigma_1}, \cdots, \boldsymbol{\Sigma_K}\}$$

General EM algorithm

Observed data: D={x1, ……., xn}

Unknown variables: y

In clustering: y=1….K clusters

Parameters: θ

In GMM : θ= π : {π1, . . . , πk}

μ : {μ1 , . . . , μK}

Σ : {Σ1, . . . ΣK }

Goal: $\hat{\theta}_n = \arg\max_{\theta} \log P(D|\theta)$

- In GMM :

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

EM algorithm for GMM

E step: Evaluate the responsibilities (posterior probabilities) using the current parameter

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)}$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

- For each example x_n,
- Compute $\gamma(z_{nk})$ the probability that x_n is generated by component Z_k i.e it belongs to cluster k
- If x_n is very likely under the k_{th} Gaussian, it gets high weight

EM algorithm for GMM

M-Step :

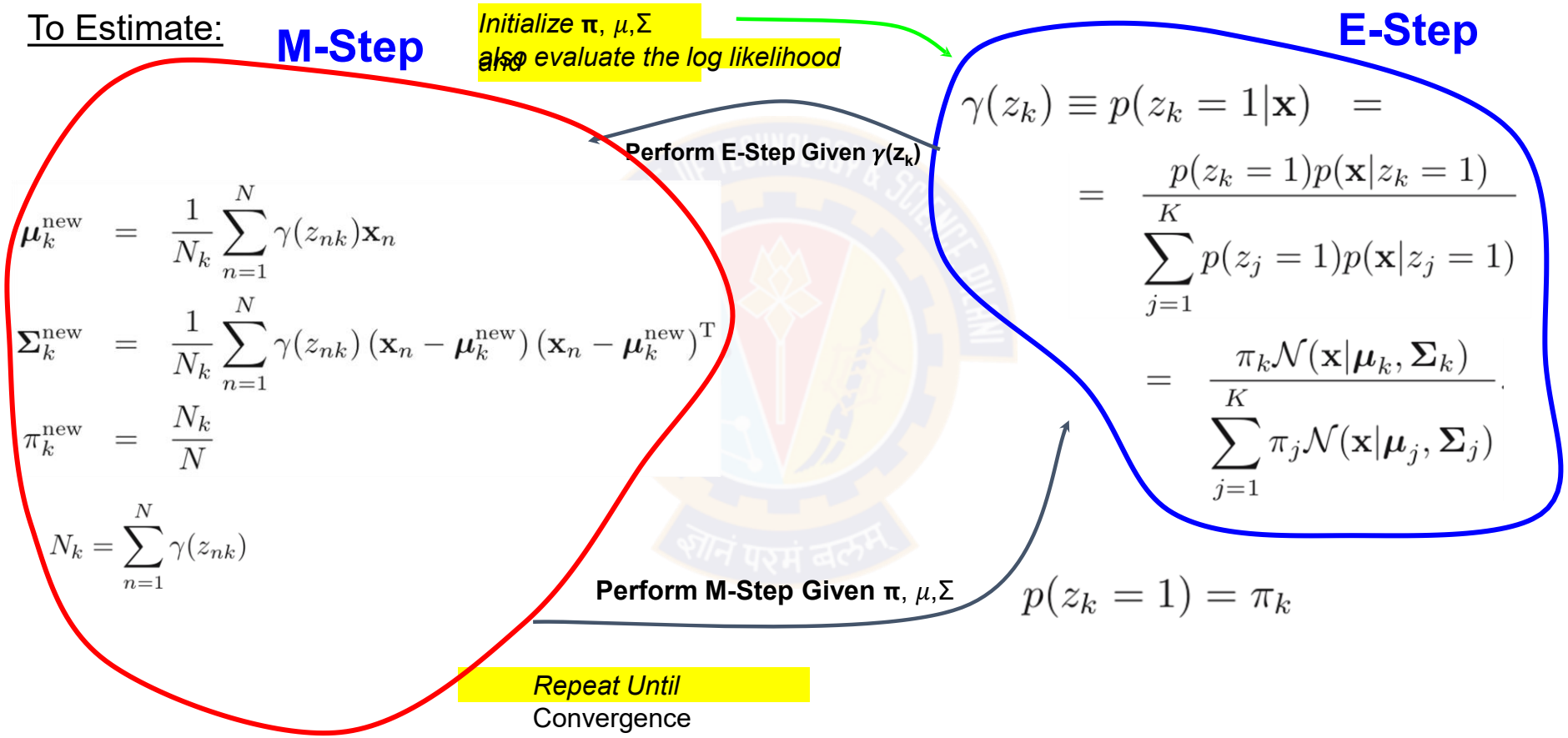
- maximize the expectation of the complete-data log-likelihood, computed with respect to the conditional probabilities found in the Expectation step. The result of the maximization is a new parameter vector μnew , Σnew and πnew
- Keep γ (znk) fixed, and apply MLE for maximizing of ln p(X|π,μ,Σ) for μk , Σk and πk, to get μnew , Σnew and πnew

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Repeat E & M until convergence

– In practice, the algorithm is deemed to have converged when the change in the log likelihood function, or alternatively in the parameters, falls below some threshold

EM algorithm for GMM


To Estimate:
M-Step
Initialize π, μ,Σ
also evaluate the log likelihood
E-Step
$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$
Perform E-Step Given $\gamma(z_k)$
$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$
$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^{\mathrm{T}}$
$\pi_k^{\text{new}} = \frac{N_k}{N}$
$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$
Perform M-Step Given π, μ,Σ
$p(z_k = 1) = \pi_k$
Repeat Until
Convergence

N_k = the effective number of points assigned to cluster k

EM Algorithm

I Standardize the data if required:

II Fix the no.of.cluster expected

III Initialize the prototypes:Mean, Covariance, Weights

IV Expectation-Step: Fix prototype & find the membership of each point weighted by the probability value

V. Calculate the log likelihood of the points

VI Maximization Step: Fix the membership(responsibility matrix) and re-estimate the prototypes

VII Calculate the new log likelihood of the points. Repeat E & M Step till convergence is achieved:

Dataset

	Word1	Word2
Doc1	2	2
Doc2	4	5
Doc3	7	2

(x, y)

	Means	SDs	Weight
Cluster 1	(3,3)	(4,0.707)	0.5
Cluster 2	(4,4)	(0.5,0.707)	0.5

GMM

	Word1	Word2
Doc1	2	2
Doc2	4	5
Doc3	7	2

	Means	SDs	Weight
Cluster 1	(3,3)	(4,0.707)	0.5
Cluster 2	(4,4)	(0.5,0.707)	0.5

IV Expectation-Step: Fix prototype & find the membership of each point weighted by the probability value

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

$N(M,\sigma) \rightarrow$ Prob. Value using Gaussian method

$$\frac{e^{-\frac{1}{2}\left[\frac{x-\mu}{\sigma}\right]^2}}{\sigma\sqrt{2\pi}}$$

Sample calculation for this step is shown below:

	Doc1	Doc2	Doc3
Weight1 * N1			
Weight2 * N2			
Sum			
P(Zi1)			
P(Zi2)			

Weight1*N1 =

$0.5 * P(Doc1, Cluter1)$

$= (Doc1, (3,3), (4,0.707))$

$$= 0.5 * \frac{e^{\frac{-(2-3)^2}{2*4^2}}}{4*\sqrt[2]{2*3.14}} * \frac{e^{\frac{-(2-3)^2}{2*0.707^2}}}{0.707*\sqrt[2]{2*3.14}}$$

δ_1 δ_2

$= 0.01004$

For Cluster 1 assumed parameters, here for Doc1 apply in the below formula and multiply with the weight = 0.5. For simplicity , assume off diagonal values in covariance matrix as 0 (This is usually done in text analysis use cases). similarly do it for all the Doc's

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

GMM

IV Expectation-Step: Fix prototype & find the membership of each point weighted by the probability value

	Word1	Word2
Doc1	2	2
Doc2	4	5
Doc3	7	2

	Means	SDs	Weight
Cluster 1	(3,3)	(4,0.707)	0.5
Cluster 2	(4,4)	(0.5,0.707)	0.5

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

	Doc1	Doc2	Doc3
Weight1 * N1			
Weight2 * N2			
Sum			
P(Zi1)			
P(Zi2)			

Similarly:

For Cluster 2 assumed parameters, here for Doc1 apply in the below formula and multiply with the weight = 0.5. For simplicity , assume off diagonal values in covariance matrix as 0(This is usually done in text analysis use cases), similarly do it for all the Doc's

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

- GMM

	Word1	Word2
Doc1	2	2
Doc2	4	5
Doc3	7	2

	Means	SDs	Weight
Cluster 1	(3,3)	(4,0.707)	0.5
Cluster 2	(4,4)	(0.5,0.707)	0.5

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

	Doc1	Doc2	Doc3
Weight1 * N1			
Weight2 * N2			
Sum			
P(Zi1)			
P(Zi2)			

Per Docs sum the previous two found values.
This gives the denominator of this formula

For Doc2 :
Weight1*N1 = 0.0005
Weight2*N2 = 0.0828
Sum = 0.08329 = ~0.0833

$P(Z_{Doc1\text{-}Cluster1})$ = 0.006 ✓
$P(Z_{Doc1\text{-}Cluster1})$ = 0.994 ✓

Now find the membership of doc :
P(Zi1) = dividing Weight1*N1 by Sum = 0.0005/0.0833 = 0.006
P(Zi2) = dividing Weight2*N1 by Sum = 0.0828/0.0833 = 0.9939

Similarly do it for all the Doc's

•GMM

	Word1	Word2
Doc1	2	2
Doc2	4	5
Doc3	7	2

	Means	SDs	Weight
Cluster 1	(3,3)	(4,0.707)	0.5
Cluster 2	(4,4)	(0.5,0.707)	0.5

V. Calculate the log likelihood of the points

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

	Doc1	Doc2	Doc3
Weight1 * N1			
Weight2 * N2			
Sum			
P(Zi1)			
P(Zi2)			

Log likehood = ln(0.010) + ln(0.083) + ln(0.006) =-12.16

Log likelihood = ln(sum of Doc1) + ln(sum of Doc2 + ln(Sum of Doc3)….

This value needs to be found for every iteration and at one iteration it will approach a very large value . That will be one of the stopping convergence point of the algorithm.

- GMM

VI Maximization Step: Fix the membership(responsibility matrix) and re-estimate the prototypes

	Word1	Word2
Doc1	2	2
Doc2	4	5
Doc3	7	2

	Means	SDs	Weight
Cluster 1	(3,3)	(4,0.707)	0.5
Cluster 2	(4,4)	(0.5,0.707)	0.5

	Doc1	Doc2	Doc3
P(Zi1)	0.9999	0.006	1
P(Zi2)	0.0001	0.994	0

	Means	SDs	Weight
Cluster 1			
Cluster 2			

Sample calculations shown below

$N_{k=1}$ = Cluster 1 =
= 0.9999+0.006+1
=2.0059
This is the value N1 represented here

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

N1 for cluster k=1 is got by summing these.
Similarly do it for N2

N = 3 (no. of data points)

For cluster k=1 is Means obtained by multiplying the data by the membership.
[0.9999(2) + 0.006(4) + 1(7)]/N1 , [0.9999(2) + 0.006(5) + 1(2)]/N1

MEAN$_{k=1}$ =
(4.49, 2.00) = ~(4.49,2)

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

Similarly do it for cluster 2

Std.DeV$_{k=1}$ =
($[0.9999(2-4.49)^2 + 0.006(4-4.49)^2 + 1(7-4.49)^2]/N1$,
$[0.9999(2-2)^2 + 0.006(5-2)^2 + 1(2-2)^2]/N1$)

= (6.23, 0.03)

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^{\mathrm{T}}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

Weight$_{k=cluster1\ (new)}$
= 2.0059/3 = 0.67

GMM - EM Algorithm

- 2nd Iteration continues

	Word1	Word2
Doc1	2	2
Doc2	4	5
Doc3	7	2

I Standardize the data if required:

II Fix the no.of.cluster expected

III Initialize the prototypes: Mean, Covariance, Weights

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

	Means	SDs	Weight
Cluster 1	(4.49,2)	(6.32,0.03)	0.67
Cluster 2	(3.99,4.99)	(0.001,0.001)	0.33

IV Expectation-Step: Fix prototype & find the membership of each point weighted by the probability value

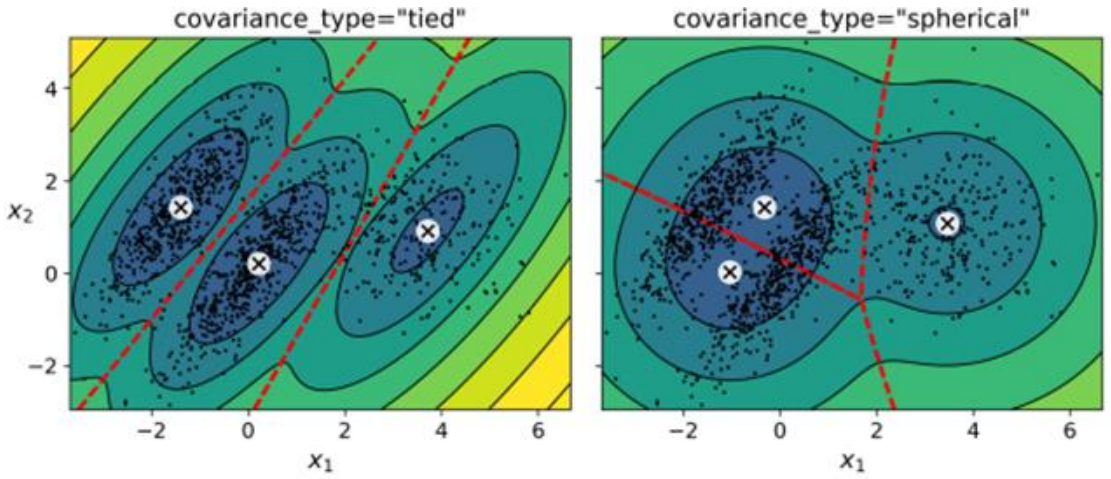
	Doc1	Doc2	Doc3
Weight1 * N1			
Weight2 * N2			
Sum			
P(Zi1)			
P(Zi2)			

V. Calculate the log likelihood of the points

VI Maximization Step: Fix the membership(responsibility matrix) and re-estimate the prototypes

VII Calculate the new log likelihood of the points. Repeat E & M Step till convergence is achieved:

Gaussian Mixture Model



Note:
The GMM model can be relaxed by setting these parameters while calling the function

- "**spherical**": all clusters must be spherical, but they can have different diameters (i.e., different variances).
- "**diag**": clusters can take on any ellipsoidal shape of any size, but the ellipsoid's axes must be parallel to the coordinate axes (i.e., the covariance matrices must be diagonal).
- "**tied**": all clusters must have the same ellipsoidal shape, size and orientation (i.e., all clusters share the same covariance matrix).

Gaussian Mixture Model

```
from sklearn.mixture import GaussianMixture

gm = GaussianMixture(n_components=3, n_init=10)
gm.fit(X)

gm.weights_
gm.means_
gm.covariances_
```

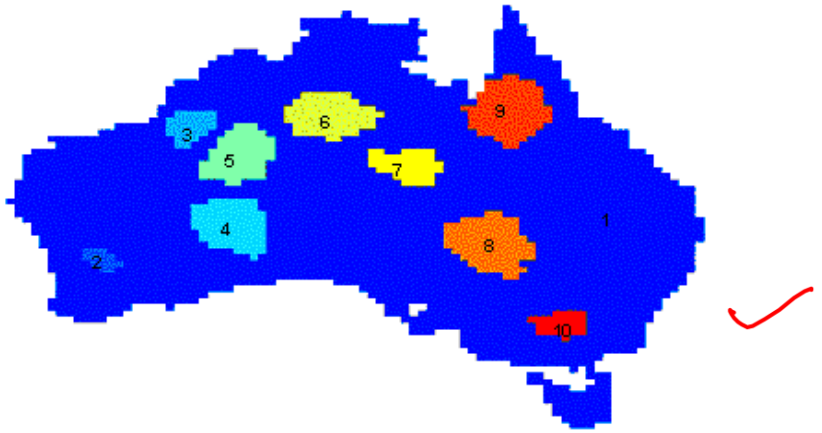
```
>>> gm.predict(X)
array([2, 2, 1, ..., 0, 0, 0])
>>> gm.predict_proba(X)
array([[2.32389467e-02, 6.77397850e-07, 9.76760376e-01],
       [1.64685609e-02, 6.75361303e-04, 9.82856078e-01],
```

Applications of Cluster Analysis

- **Understanding**
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

- **Summarization**
  - Reduce the size of large data sets



- Clustering precipitation in Australia

What is not Cluster Analysis?

- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification
- Graph partitioning
  - Some mutual relevance and synergy, but areas are not identical

Types of Clustering

- A **clustering** is a set of clusters
- **Partitional Clustering**
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- **Hierarchical clustering**
  - A set of nested clusters organized as a hierarchical tree

k means clustering – numerical example

Consider the following points: A(1,2),B(1,4),C(1,0),D(10,2),E(10,4),F(10,0).Perform k means clustering with k = 2.

- **We randomly pick 2 initial centroids. Let's say:**

  Centroid 1: (1, 2) — same as Point A

  Centroid 2: (10, 2) — same as Point D

- **Assign Each Point to the Nearest Centroid**

Point	Dist.C1(1,2)	Dist.C2(10,2)	Cluster
A(1,2)	0	9	
B(1,4)	2	9.22	
C(1,0)	2	9.22	
D(10,2)	9	0	
E(10,4)	9.22	2	
F(10,0)	9.22	2	

k means clustering – numerical example

Point	Dist.C1(1,2)	Dist.C2(10,2)	Cluster
A(1,2)	0	9	1
B(1,4)	2	9.22	1
C(1,0)	2	9.22	1
D(10,2)	9	0	2
E(10,4)	9.22	2	2
F(10,0)	9.22	2	2

- Compute New Centroids
  - Cluster 1 contains A (1,2), B (1,4), C (1,0)

    New centroid = (1+1+1/3,2+4+0)/3) → **(1, 2)**
  - Cluster 2 contains D (10,2), E (10,4), F (10,0)

    New centroid = (10+10+10/3,2+4+0)/3) = 2 → **(10, 2)**

- The centroids didn't change, so the algorithm **converges**.
- **Cluster 1**: A, B, C
- **Cluster 1**: D, E, F

k means clustering – another example

Question : Consider the following points: A(2,10),B(2,5),C(8,4),D(5,8),E(7,5),F(6,4).Perform k means clustering with k = 2

- Let us choose centroids randomly as

Centroid 1 (C1) = A = (2, 10),Centroid 2 (D) = P4 = (5, 8)

Point	Dist.C1(2,10)	Dist.C2(5,8)	Cluster
A(2,10)	0	$\sqrt{[(5-2)^2 + (8-10)^2]} = \sqrt{13} \approx 3.6$	1
B(2,5)	$\sqrt{[(2-2)^2 + (5-10)^2]} = \sqrt{25} = 5$	$\sqrt{[(5-2)^2 + (8-5)^2]} = \sqrt{18} \approx 4.2$	1
C(8,4)	$\sqrt{[(8-2)^2 + (4-10)^2]} = \sqrt{72} \approx 8.5$	$\sqrt{[(5-8)^2 + (8-4)^2]} = \sqrt{25} = 5$	2
D(5,8)	$\sqrt{[(5-2)^2 + (8-10)^2]} = \sqrt{13} \approx 3.6$	0	2
E(7,5)	$\sqrt{[(7-2)^2 + (5-10)^2]} = \sqrt{50} \approx 7.1$	$\sqrt{[(5-7)^2 + (8-5)^2]} = \sqrt{13} \approx 3.6$	2
F(6,4)	$\sqrt{[(6-2)^2 + (4-10)^2]} = \sqrt{52} \approx 7.2$	$\sqrt{[(5-6)^2 + (8-4)^2]} = \sqrt{17} \approx 4.1$	2

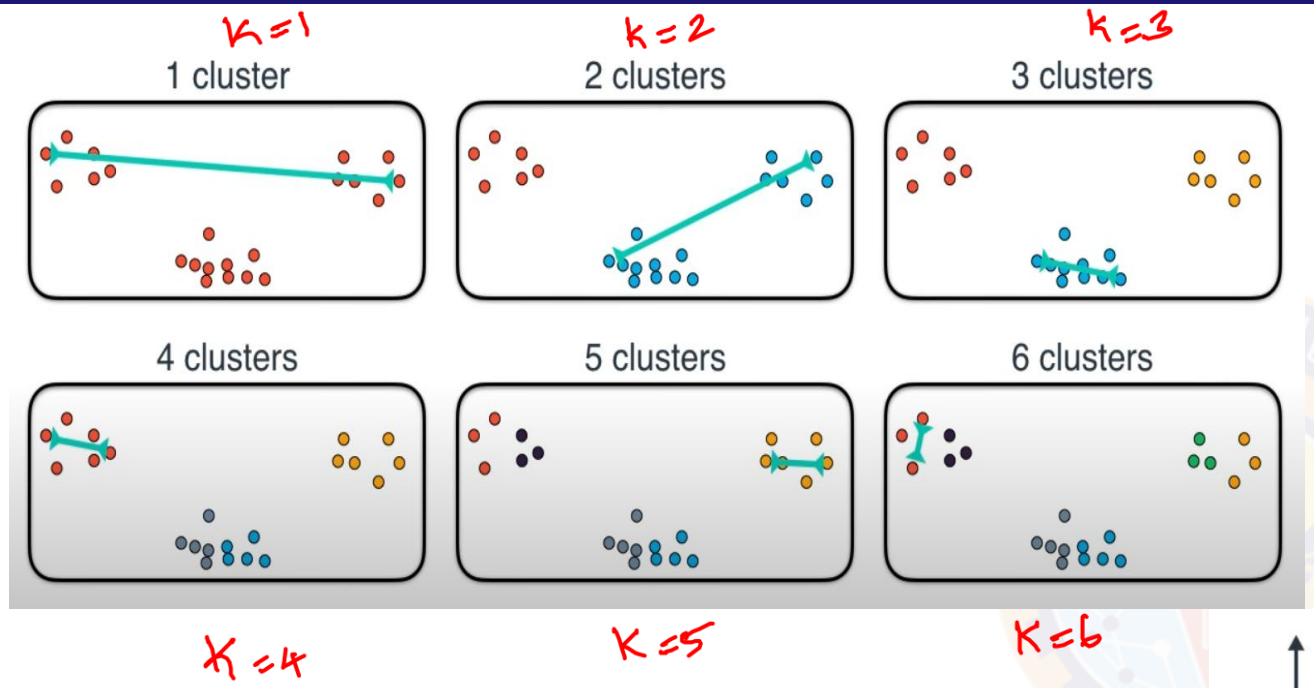
- **Compute New Centroids**
  - **Cluster 1**: A (2,10), B (2,5)
    - New centroid: [(2+2)/2 , (10+5)/2] = **(2,7)**
  - **Cluster 2**: C,D,E,F
    - New centroid (8+5+7+6)/4 , (4+8+5+4)/4 → **(6.5, 5.25)**

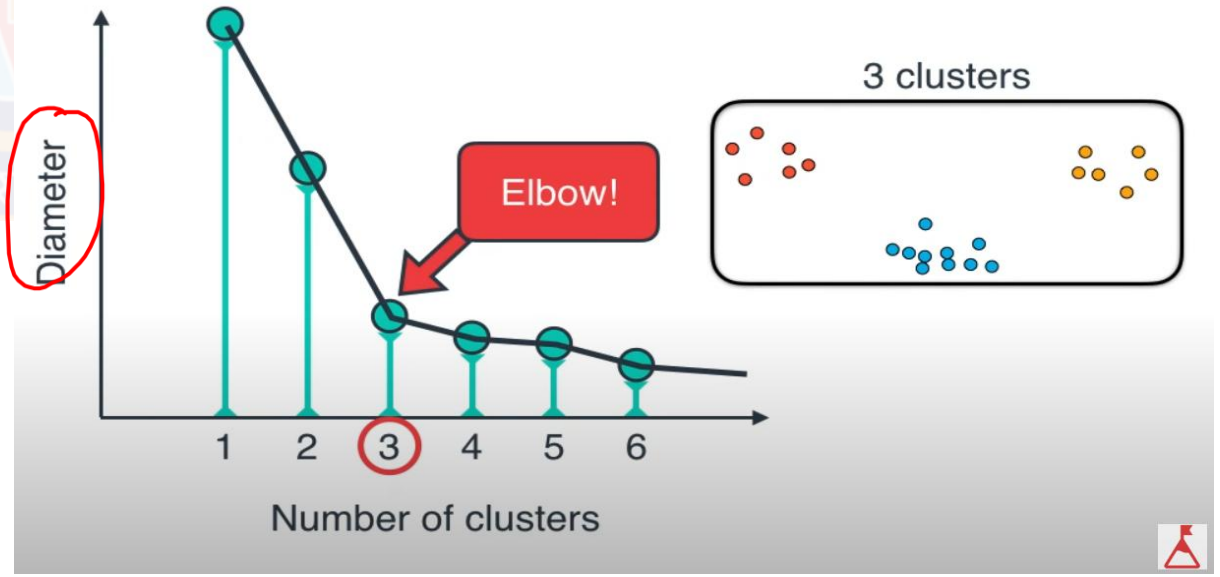
k means clustering – another example

Point	Dist.C1(2,7)	Dist.C2(6.5,5.25)	Cluster
A(2,10)	$\sqrt{[(2-2)^2 + (10-7)^2]} = 3$	$\sqrt{[(6.5-2)^2 + (5.25-10)^2]} \approx 6.7$	1
B(2,5)	$\sqrt{[(2-2)^2 + (5-7)^2]} = 2$	$\sqrt{[(6.5-2)^2 + (5.25-5)^2]} \approx 4.5$	1
C(8,4)	$\sqrt{[(8-2)^2 + (4-7)^2]} \approx 6.7$	$\sqrt{[(6.5-8)^2 + (5.25-4)^2]} \approx 1.8$	2
D(5,8)	$\sqrt{[(5-2)^2 + (8-7)^2]} \approx 3.2$	$\sqrt{[(6.5-5)^2 + (5.25-8)^2]} \approx 3.1$	2
E(7,5)	$\sqrt{[(7-2)^2 + (5-7)^2]} \approx 5.4$	$\sqrt{[(6.5-7)^2 + (5.25-5)^2]} \approx 0.5$	2
F(6,4)	$\sqrt{[(6-2)^2 + (4-7)^2]} \approx 5$	$\sqrt{[(6.5-6)^2 + (5.25-4)^2]} \approx 1.3$	2

- New Clusters (No Change) → Converged
  - Cluster 1:A,B
  - Cluster 2: C,D,E,F

Elbow Method

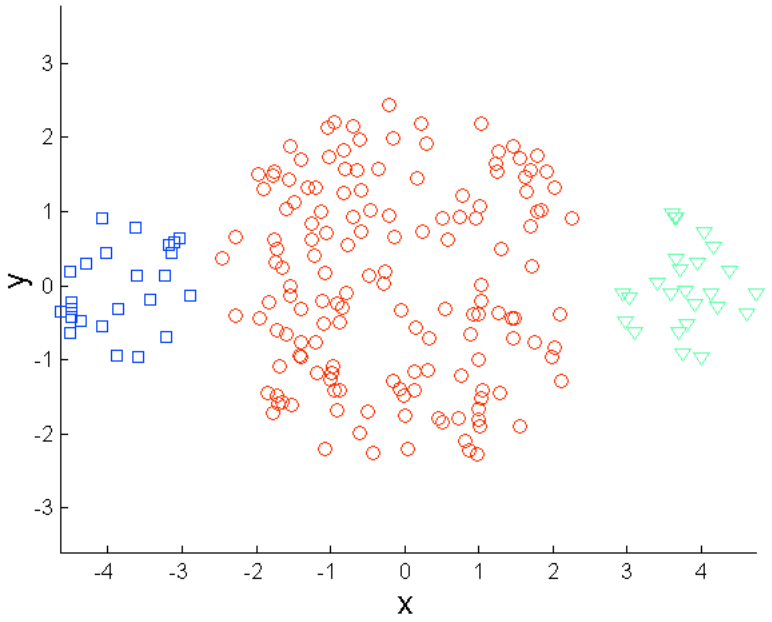




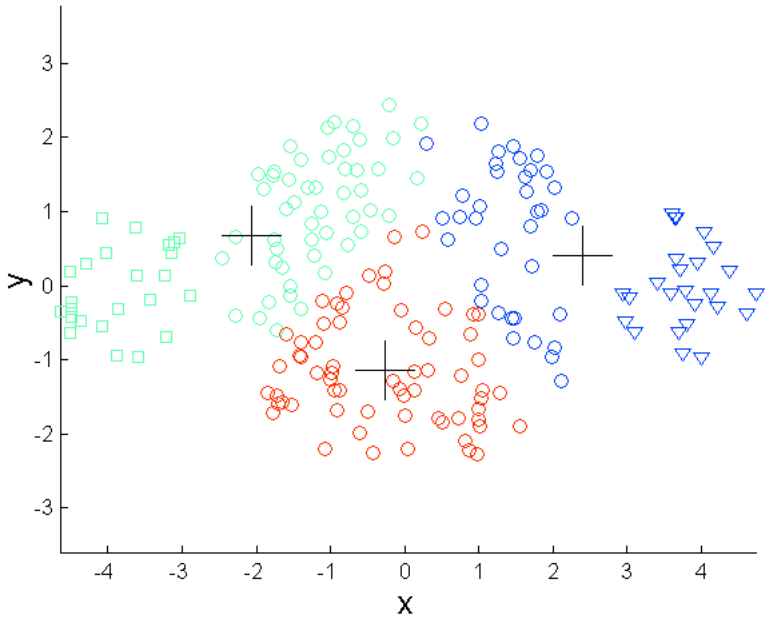
Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes



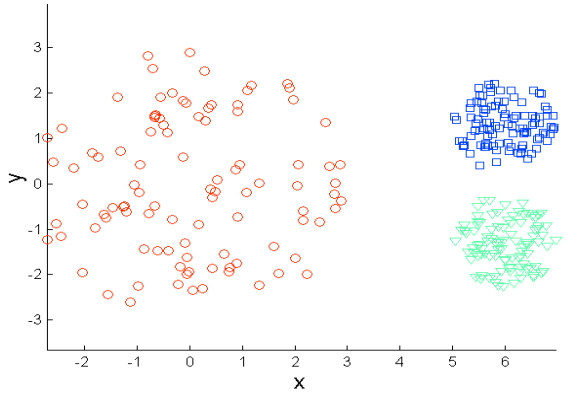
Original Points



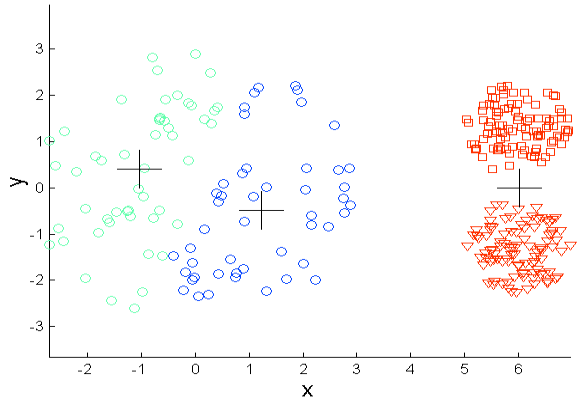
K-means (3 Clusters)

Limitations of K-means

Differing Density

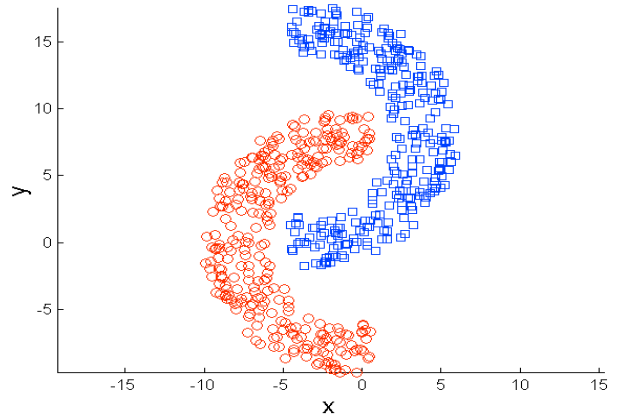


Original Points

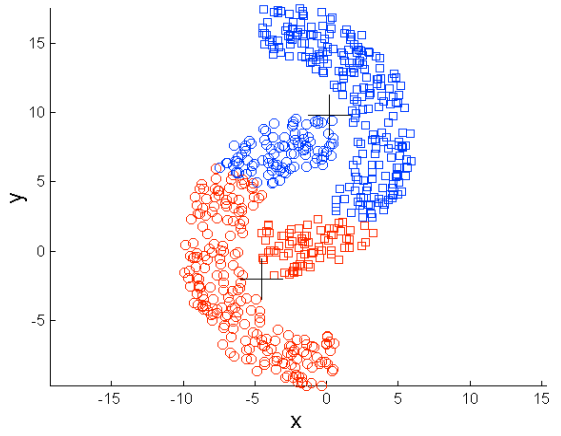


K-means (3 Clusters)

Non-globular Shapes



Original Points



K-means (2 Clusters)

Comments on the K-Means Method

- Strength: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*.
- Weakness
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

k-Medoids: A Representative Object-Based Technique

"How can we modify the k-means algorithm to diminish such sensitivity to outliers?"

- **Differences:**
  - *Medoid*: A medoid is the most centrally located object within a cluster, meaning it has the smallest average dissimilarity to other points in the cluster. Unlike the centroid (mean), the medoid is always one of the actual data points.
  - *Dissimilarity*: Instead of using squared distances (as in k-Means), k-Medoids can work with a variety of distance measures (e.g., Euclidean, Manhattan) or even dissimilarity matrices, making it more flexible for non-numerical data.

The K-Medoid Clustering Method

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
    - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* : PAM on samples
  - *CLARANS* : Randomized re-sampling

Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall

- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
    - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

➢ For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity

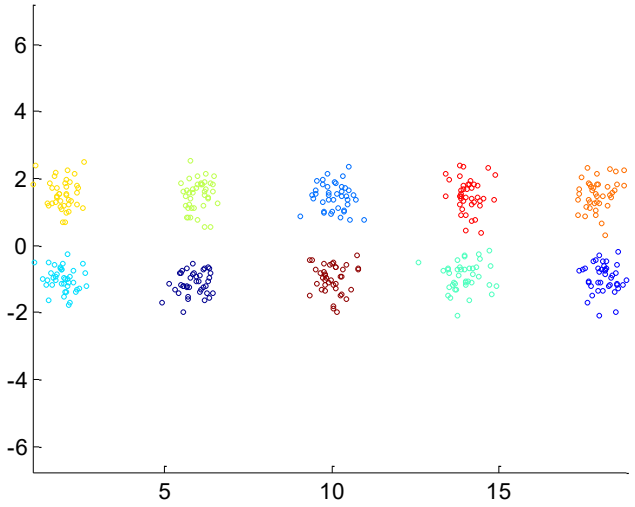
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as criteria instead of indices
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

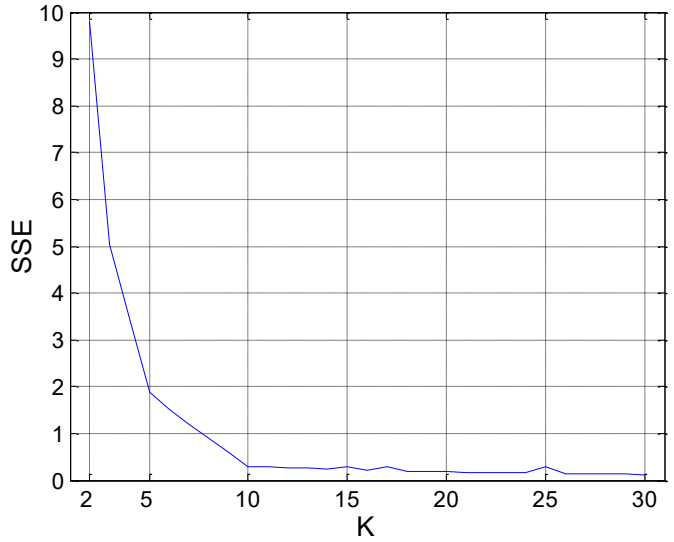
Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix
  - “Incidence” Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density based clusters.

Internal Measures: SSE

- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters





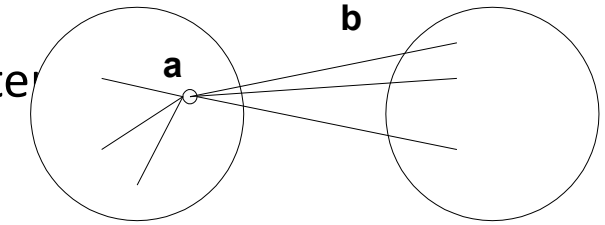
Framework for Cluster Validity

- Need a framework to interpret any measure.
  - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity
  - The more “atypical” a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
    - If the value of the index is unlikely, then the cluster results are valid
  - These approaches are more complicated and harder to understand.
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
  - However, there is the question of whether the difference between two index values is significant

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by

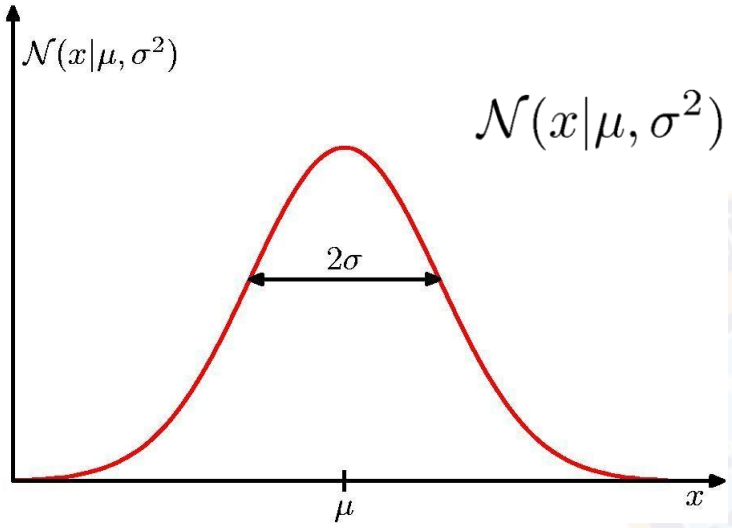
    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)
  - Typically between 0 and 1.
  - The closer to 1 the better.
- Can calculate the Average Silhouette width for a cluster or a clustering



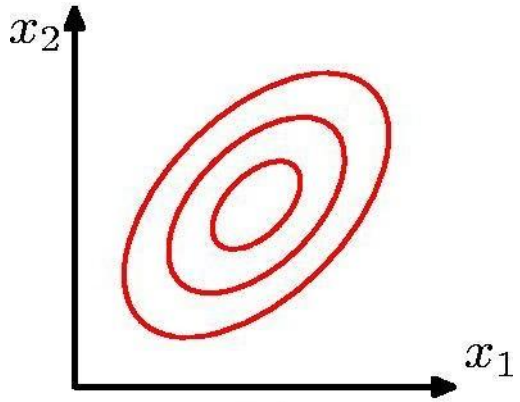
Probabilistic Clustering

- Represent the probability distribution of the data as a *mixture model*
  - captures uncertainty in cluster assignments
  - gives model for data distribution
  - *Bayesian* mixture model allows us to determine *K*
- Consider mixtures of *Gaussians*

Review: Gaussian Distribution



$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Mixtures of Gaussians

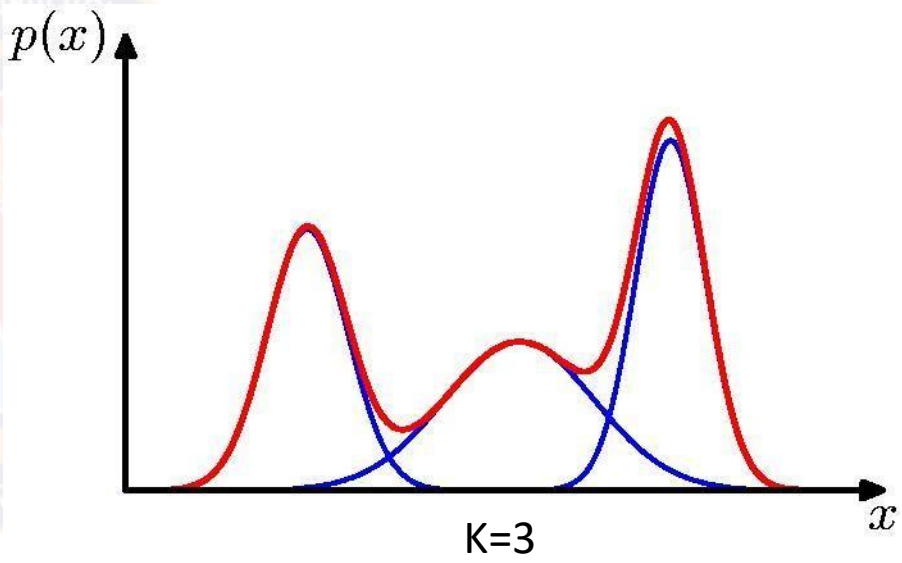
- Combine simple models into a complex model:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Component

Mixing coefficient

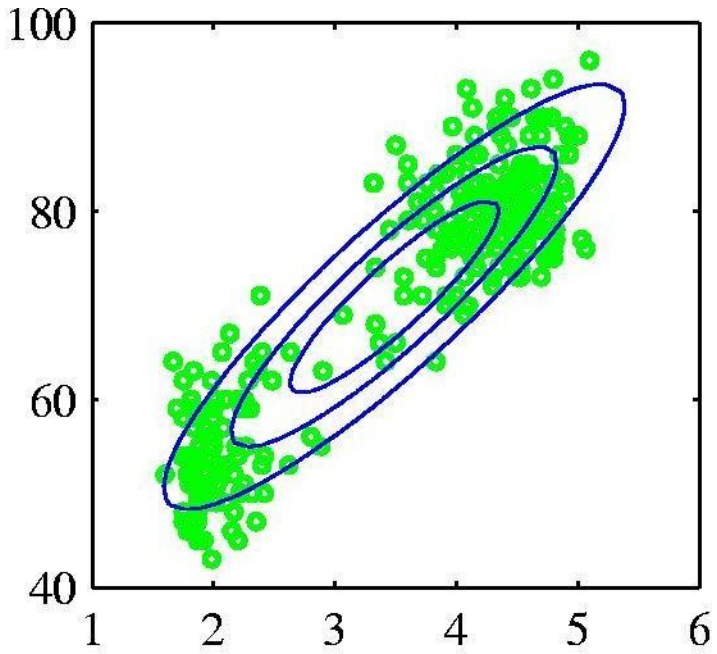
$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$



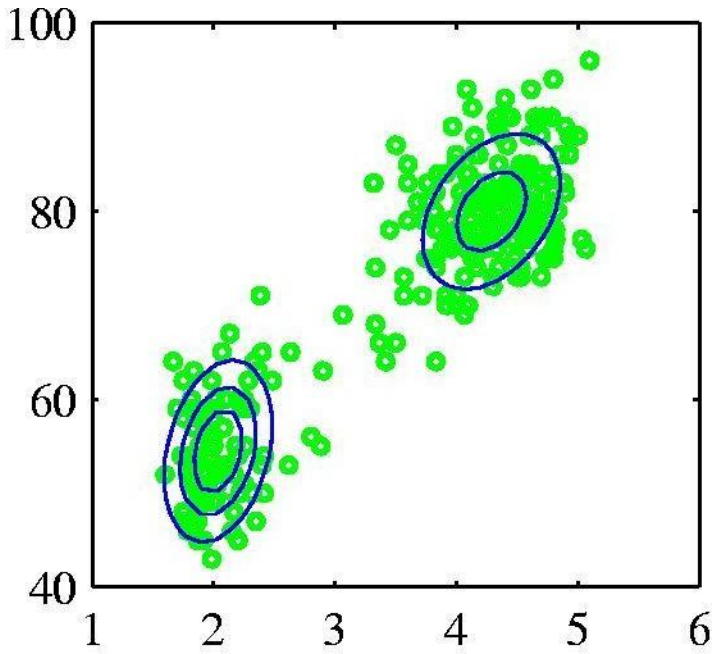
K=3

- Find parameters through EM (Expectation Maximization) algorithm

Probabilistic version: Mixtures of Gaussians



Single Gaussian



Mixture of two Gaussians

Thank You!